

# **PENGEMBANGAN APLIKASI *TRAFFIC LIGHT* E-TILANG MENGUNAKAN GOOGLE GEOFENCING API BERBASIS ANDROID**

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Ditya Enandini Palupi  
NIM: 145150201111080



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PENGEMBANGAN APLIKASI TRAFFIC LIGHT MENGGUNAKAN GOOGLE  
GEOFENCING API BERBASIS ANDROID

SKRIPSI


Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer


Disusun Oleh :  
Ditya Enandini Palupi  
NIM: 145150201111080

Skripsi ini telah diuji dan dinyatakan lulus pada  
2 Agustus 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I


Dosen Pembimbing II

  
Muhammad Aminul Akbar, S.Kom., M.T.  
NIK: 2016078910131001

  
Adam Hendra Brata S.Kom., M.T., M.Sc  
NIK: 2016079001051001

Mengetahui  
Ketua Jurusan Teknik Informatika



  
Tri Astoro Kurniawan, S.T., M.T., Ph.D.  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Agustus 2018



Ditya Enandini Palupi

NIM: 145150201111080

## KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan hidayah-Nya yang memberi kemudahan sehingga penulis dapat menyelesaikan skripsi yang berjudul “PENGEMBANGAN APLIKASI TRAFFIC LIGHT E-TILANG MENGGUNAKAN GOOGLE GEOFENCING API BERBASIS ANDROID”.

Untuk kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak – pihak yang telah membantu penulis selama penyusunan skripsi, diantaranya:

1. Bapak Muhammad Aminul Akbar, S.Kom., M.T selaku dosen pembimbing I yang telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
2. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc selaku dosen pembimbing II yang juga telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Drs. Mardji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widowo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
6. Kepada kedua orang tua penulis, yaitu Bapak Didik Sudiwinarto dan Ibu Tri Widyaningrum beserta keluarga besar yang selalu memberikan segala masukan, do’a, motivasi dan semangat yang tidak terputus.
7. Muhammad Yudha Yusi Putra yang telah membantu dan memberikan

semangat setiap harinya dalam penyelesaian Skripsi ini.

8. Rekan-rekan LBB Anton Firdaus yang membantu memberikan ilmu dalam implementasi dari penelitian ini.
9. Teman-teman Kopma Squad yang selalu mengisi hari-hari menjadi menyenangkan.
10. Sahabat-sahabatku Alifah Roscahyaning, Rizka Syah, Edo Brastagi, Fitri Diah, dan Nidya Elvira. Terima kasih atas dorongan semangat dan kebersamaan yang tidak terlupakan.
11. Semua pihak yang tidak semuanya bisa dituliskan disini yang terlibat secara langsung maupun tidak langsung dalam proses pengerjaan skripsi maupun sebagai pemberi semangat dan motivasi.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan penelitian skripsi ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Malang, 2 Agustus 2018

Penulis

enandiniditya@yahoo.com



## ABSTRAK

Tingkat pelanggaran lalu lintas di Indonesia semakin meningkat. Hal ini terbukti karena meningkatnya angka kecelakaan yang ada di kota Surabaya. Dalam rangka mengurangi penurunan pelanggaran lalu lintas jalan, maka diperlukan aturan atau pembinaan kepada masyarakat untuk pentingnya menaati peraturan lalu lintas yang sudah ditetapkan oleh pemerintah. Dinas Perhubungan Kota Surabaya merupakan salah satu lembaga atau instansi yang menyediakan prasarana dan sarana transportasi untuk mendukung kegiatan pembangunan daerah serta meningkatkan kesejahteraan masyarakat, salah satunya dengan memfasilitasi kamera CCTV. Kamera tersebut difungsikan untuk menerapkan sistem tilang elektronik yang beroperasi sejak awal September 2017. Dengan adanya bantuan CCTV tersebut, sistem dapat membantu mengidentifikasi jenis pelanggaran, pelaku, hingga lokasi pelanggaran berada. Penulis melakukan survei kepada 30 responden terkait kebutuhan-kebutuhan yang dibutuhkan untuk pengguna kendaraan yang belum mengetahui kapan serta dimana melakukan pelanggaran tersebut. Dengan latar belakang tersebut dibuatlah sebuah aplikasi sistem pemantau *traffic light* untuk mempermudah pengguna kendaraan dalam menemukan informasi pelanggaran dan memberikan pengguna sebuah peringatan yang memasuki daerah *traffic light*. Implementasi dilakukan dengan Android Studio. Pengujian dilakukan dengan melewati suatu daerah dan hasil yang didapatkan yaitu hasil jarak berdasarkan kecepatan kendaraan.

Kata kunci: *traffic light*, Pengguna, Notifikasi

## ABSTRACT

The level of traffic violations in Indonesia is increasing. This is evidenced by the increasing number of accidents in the city of Surabaya. In order to reduce the reduction of road traffic violations, it is necessary to have rules or guidance to the public for the importance of obeying the traffic rules set by the government. Surabaya Transportation Department is one of institutions or agencies that provide infrastructure and transportation facilities to support local development activities and improve the welfare of the community, one of them by facilitating CCTV cameras. The camera is enabled to implement an electronic ticketing system that operates since the beginning of September 2017. With the help of CCTV, the system can help identify the types of violations, offenders, to where the offense is located. The authors conducted a survey to 30 respondents terkait necessary needs for vehicle users who do not know when and where the conduct of the offense. Against this background, a traffic light monitoring system application was created to make it easier for vehicle users to locate infringement information and give users a warning that enters the traffic light area. Implementation is done with Android Studio. Testing is done by passing an area and the result of distance based on vehicle speed.

Keywords: *Traffic Light*, Users, Notifications

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN .....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Dinas Perhubungan.....	5
2.3 <i>Geofencing API</i> .....	6
2.4 Android Studio .....	6
2.5 <i>Software Development Life Cycle</i> .....	7
2.5.1 <i>Model Waterfall</i> .....	8
BAB 3 METODOLOGI .....	10
3.1 Studi Pustaka.....	10
3.2 Pengumpulan Data .....	11
3.3 Analisis Kebutuhan .....	11
3.4 Perancangan .....	11
3.5 Implementasi .....	11
3.6 Pengujian .....	12



3.7 Kesimpulan dan Saran .....	12
BAB 4 ANALISIS KEBUTUHAN .....	13
4.1 Gambaran Umum Sistem.....	13
4.2 Analisis Data.....	13
4.3 Identifikasi Aktor.....	14
4.4 Kebutuhan Fungsional Sistem .....	14
4.5 Diagram <i>Use Case</i> .....	15
4.6 Skenario <i>Use Case</i> .....	16
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	19
5.1 Perancangan .....	19
5.1.1 Perancangan Arsitektur Sistem.....	19
5.1.2 Perancangan <i>Sequence Diagram</i> .....	20
5.1.3 Perancangan <i>Class Diagram</i> .....	21
5.1.4 Perancangan Basis Data .....	22
5.1.5 Perancangan Antarmuka.....	23
5.1.6 Perancangan Algoritme.....	29
5.2 Implementasi .....	30
5.2.1 Spesifikasi Sistem .....	30
5.2.2 Implementasi Algoritma.....	31
5.2.3 Implementasi Antarmuka .....	34
BAB 6 Pengujian .....	39
6.1 Pengujian Validasi .....	39
6.2 Pengujian Performa .....	45
6.3 Analisa Hasil Pengujian .....	45
6.3.1 Pengujian Validasi .....	46
6.3.2 Pengujian Performa .....	46
BAB 7 Penutup .....	47
7.1 Kesimpulan.....	47
7.2 Saran .....	47
DAFTAR PUSTAKA.....	48
LAMPIRAN A KUISIONER PENGGUNA .....	50
lampiran B data pelanggaran .....	55

## DAFTAR TABEL

Tabel 4.1 Analisis Data Kebutuhan .....	13
Tabel 4.2 Identifikasi Aktor .....	14
Tabel 4.3 Kebutuhan Fungsional Sistem .....	14
Tabel 4.4 Kebutuhan Fungsional Sistem (Lanjutan).....	15
Tabel 4.5 Kebutuhan Non Fungsional Sistem .....	15
Tabel 4.6 Skenario <i>Use Case Login</i> .....	16
Tabel 4.7 Skenario <i>Use Case</i> Melihat Informasi Pelanggaran dan Estimasi Denda .....	16
Tabel 4.8 Skenario <i>Use Case</i> Melihat Notifikasi Pemberitahuan Mendekati <i>Traffic Light</i> .....	17
Tabel 4.9 Skenario <i>Use Case</i> Melihat Layanan Keadaan Darurat .....	17
Tabel 5.1 Perancangan Basis Data Tabel Informasi .....	23
Tabel 5.2 Penjelasan Antarmuka Menu Utama .....	24
Tabel 5.3 Penjelasan Antarmuka <i>Input</i> Nomor Polisi .....	25
Tabel 5.4 Penjelasan Antarmuka Pindai Kode .....	26
Tabel 5.5 Penjelasan Antarmuka Informasi Pelanggaran .....	27
Tabel 5.6 Penjelasan Antarmuka Panggilan Darurat .....	28
Tabel 5.7 Perancangan Algoritme Login .....	29
Tabel 5.8 Perancangan Algoritme Notifikasi.....	29
Tabel 5.9 Spesifikasi Perangkat Keras Komputer .....	30
Tabel 5.10 Spesifikasi Perangkat Keras <i>Smartphone</i> .....	30
Tabel 5.11 Spesifikasi Perangkat Lunak Komputer .....	31
Tabel 5.12 Spesifikasi Perangkat Lunak <i>Smartphone</i> .....	31
Tabel 5.13 Implementasi Algoritma Login .....	31
Tabel 5.14 Implementasi Algoritma Notifikasi .....	32
Tabel 6.1 Kasus Uji Memasukkan Nomor Polisi Kendaraan.....	39
Tabel 6.2 Melihat Informasi Pelanggaran Dan Estimasi Denda .....	40
Tabel 6.3 Melihat Notifikasi Pemberitahuan Mendekati <i>Traffic Light</i> .....	40
Tabel 6.4 Melihat Layanan Keadaan Darurat.....	41
Tabel 6.5 Hasil Pengujian Validasi.....	42

Tabel 6.6 Pengujian Performa.....	45
-----------------------------------	----



## DAFTAR GAMBAR

Gambar 2.1 Konsep Dasar Geofence .....	6
Gambar 2.2 Model <i>Waterfall</i> (Sumber: Sommerville (2011)) .....	8
Gambar 3.1 Metode Penelitian.....	10
Gambar 4.1 Diagram <i>Use Case</i> Sistem .....	15
Gambar 5.1 Perancangan Arsitektur Sistem .....	19
Gambar 5.2 <i>Sequence Diagram</i> Login .....	20
Gambar 5.3 <i>Sequence Diagram</i> Layanan Darurat.....	21
Gambar 5.4 Perancangan <i>Class Diagram</i> .....	22
Gambar 5.5 Perancangan <i>Class Diagram</i> .....	22
Gambar 5.6 Perancangan Antarmuka Menu Utama .....	24
Gambar 5.7 Perancangan Antarmuka <i>Input</i> Nomor Polisi.....	25
Gambar 5.8 Perancangan Antarmuka Pindai Kode.....	26
Gambar 5.9 Perancangan Antarmuka Informasi Pelanggaran .....	27
Gambar 5.10 Perancangan Antarmuka Panggilan Darurat.....	28
Gambar 5.11 Antarmuka Menu Utama .....	34
Gambar 5.12 Antarmuka <i>Input</i> Nomor Polisi.....	35
Gambar 5.13 Antarmuka Pindai Kode.....	36
Gambar 5.14 Antarmuka Informasi Pelanggaran .....	37
Gambar 5.15 Antarmuka Panggilan Darurat.....	38

## DAFTAR LAMPIRAN

LAMPIRAN A KUISIONER PENGGUNA .....	50
LAMPIRAN B DATA PELANGGARAN .....	55



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Tingkat pelanggaran lalu lintas di Indonesia semakin meningkat. Hal ini terbukti karena meningkatnya angka kecelakaan yang ada di kota besar (Mohammad Ridwan, 2017). Tingkat jumlah kendaraan yang beredar dari tahun ketahun semakin meningkat. Menurut Raydian Kokroso selaku Kasatlantas Polrestabes Surabaya AKBP, hal ini dapat memicu pengaruh keamanan lalu lintas yang semakin sering terjadi dan pelanggaran lalu lintas yang menimbulkan kecelakaan lalu lintas serta kemacetan lalu lintas (Wahyudin, 2014).

Dalam rangka mengurangi penurunan pelanggaran lalu lintas jalan, maka diperlukan aturan atau pembinaan kepada masyarakat untuk pentingnya menaati peraturan lalu lintas yang sudah ditetapkan oleh pemerintah khususnya Kepolisian dan Dinas Perhubungan. Dinas Perhubungan Kota Surabaya merupakan salah satu lembaga atau instansi yang menyediakan prasarana dan sarana transportasi untuk mendukung kegiatan pembangunan daerah serta meningkatkan kesejahteraan masyarakat, salah satunya dengan memfasilitasi kamera CCTV. Kamera tersebut di fungsikan untuk menerapkan sistem tilang elektronik yang beroperasi sejak awal September 2017. Dengan di terapkannya sistem tersebut, tilang CCTV ini mampu mengurangi angka pelanggaran lalu lintas hingga lebih dari 50% (Syaikhul Hadi, 2017).

Untuk mengatasi permasalahan tersebut maka, kota Surabaya mempunyai sistem CCTV yang dapat membantu mengidentifikasi jenis pelanggaran, pelaku, hingga lokasi pelanggaran berada. Setelah mendata pelaku tersebut, petugas Dinas Perhubungan Kota Surabaya akan meneruskannya ke bagian penegakan hukum. Namun, para pelaku pelanggar lalu lintas belum mengetahui kapan serta dimana melakukan pelanggaran tersebut. Para pelaku pelanggar lalu lintas baru akan mengetahui pelanggarannya ketika membayar pajak kendaraan tersebut. Jenis pelanggaran yang terekam oleh CCTV yaitu kejadian dimana pelanggar menerobos lampu merah, melanggar marka jalan, dan salah jalur.

Salah satu penyebab terjadinya pelanggaran lalu lintas adalah kurangnya waspada para pengguna kendaraan ketika hendak melewati *traffic light*. Pada umumnya, pengguna kendaraan bermotor sering menggunakan kecepatan yang tinggi sehingga menyebabkan terjadinya kecelakaan dan secara tidak langsung perilaku tersebut melanggar peraturan lalu lintas. Maka dari itu, perlunya sebuah aplikasi yang mampu memberikan peringatan dini kepada pengguna untuk mengurangi kecepatan ketika hendak melewati *traffic light* dan memberikan informasi secepat mungkin agar para pelanggar mengetahui jenis pelanggaran dan berusaha tidak mengulanginya kembali. Dengan bantuan *Google Geofencing API*, maka suatu aplikasi dapat memulai interaksi dialog dengan pengguna jika perangkat *mobile* memasuki atau keluar dari area yang telah ditentukan (Garzon & Deva, 2014).



Menurut laporan kuartal II yang disusun biro marketing bernama Waiwai Marketing, secara umum jumlah penggunaan ponsel android di Indonesia sebanyak 41 juta pengguna atau pangsa pasarnya 94%. Sementara iOS di Indonesia hanya digunakan 2,8 juta pengguna atau 6% (Adi Filda, 2015). Dan berdasarkan hasil survei yang telah dilakukan, sebanyak 95,7 persen responden mengatakan perlu adanya aplikasi untuk mengetahui jenis pelanggaran, waktu pelanggaran, dan lokasi pelanggaran yang telah dilanggar berbasis *smartphone*.

Berdasarkan permasalahan yang telah dijelaskan tersebut, akan dilakukan penelitian skripsi di Dinas Perhubungan Kota Surabaya dengan mengangkat judul “Pengembangan Aplikasi *Traffic Light* E-Tilang Menggunakan Google Geofencing API Berbasis Android” dengan menggunakan sistem android yang saat ini sudah banyak dipakai oleh masyarakat Indonesia, akan sangat mudah untuk membantu serta memudahkan masyarakat dalam mengetahui jenis pelanggaran, waktu pelanggaran serta lokasi pelanggaran yang telah dilanggar secara mudah diakses dengan menggunakan aplikasi yang ada di *smartphone* berupa android.

## 1.2 Rumusan masalah

Dari uraian latar belakang diatas, maka yang menjadi rumusan masalah adalah :

1. Bagaimana mengidentifikasi persyaratan fungsional maupun non fungsional dari aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android ?
2. Bagaimana merancang aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android ?
3. Bagaimana mengimplementasikan aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android dengan menggunakan pemodelan berorientasi obyek ?
4. Bagaimana hasil dan tingkat akurasi aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android ?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah :

1. Mengidentifikasi persyaratan fungsional maupun non fungsional dari aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android.
2. Merancang aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android.
3. Mengimplementasikan aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android dengan menggunakan pemodelan berorientasi obyek.
4. Untuk menguji tingkat akurasi aplikasi *traffic light* e-tilang menggunakan Google Geofencing API berbasis android.

## 1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut :

1. Membantu masyarakat untuk mengetahui pelanggaran lalu lintas yang telah dilanggar.
2. Pengimplementasian sistem yang diperoleh dari penelitian ini dapat digunakan untuk mengurangi tingkat kecelakaan kedepan di Kota Surabaya.

## 1.5 Batasan masalah

Batasan masalah dalam “Pengembangan Aplikasi Traffic Light E-Tilang Menggunakan Google Geofencing API Berbasis Android” adalah :

1. Aplikasi ini dibuat hanya untuk data pelanggar lalu lintas yang ada di Kota Surabaya.
2. Data yang digunakan berasal dari Dinas Perhubungan Kota Surabaya.
3. Aplikasi ini hanya kompatibel dengan ponsel berbasis Android minimal 6.0 (Marshmallow).

## 1.6 Sistematika pembahasan

### BAB I PENDAHULUAN

Bab ini berisi latar belakang yang menjadi dasar penelitian, rumusan masalah yang berkaitan dengan hal yang ingin dikaji dalam penelitian, tujuan yang ingin didapatkan dari penelitian, manfaat dari penelitian yang sudah dilaksanakan, batasan masalah, serta sistematika pembahasan.

### BAB II TINJAUAN PUSTAKA

Bagian tinjauan pustaka menguraikan teori penunjang yang mendasari proses penelitian.

### BAB III METODOLOGI PENELITIAN

Bab metode penelitian berisi penjelasan mengenai studi literatur sebagai acuan dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta penarikan kesimpulan dan saran.

### BAB IV REKAYASA KEBUTUHAN

Dalam bab ini diuraikan penjelasan mengenai analisis kebutuhan yang dibutuhkan untuk membangun aplikasi sesuai dengan apa yang dibutuhkan baik kebutuhan baik pengguna maupun kebutuhan bagi sistem yang akan dibangun.

### BAB V PERANCANGAN DAN IMPLEMENTASI

Bagian ini berisi deskripsi perancangan struktur, komponen, representasi data, serta implementasi dari fungsi yang menjadi fokus yang didasarkan pada apa yang telah dirancang sebelumnya.

## **BAB VI PENGUJIAN**

Pada bab ini menjelaskan mengenai strategi dan teknik pengujian yang dilakukan serta penjelasan mengenai analisis yang telah dilakukan terhadap sistem yang dibangun, dan terdapat kasus uji serta hasil pengujian yang didapatkan.

## **BAB VII PENUTUP**

Dalam bagian penutup menjelaskan mengenai kesimpulan yang berisikan jawaban dari rumusan masalah yang telah dibuat sebelumnya dengan berdasarkan hasil penelitian, dan terdapat saran guna mengembangkan penelitian lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Terdapat beberapa penelitian terdahulu yang dijadikan referensi untuk mendukung penelitian ini. Penelitian pertama yang dilakukan oleh Halim Wongsokuncoro (2016) dengan judul “Rancang Bangun Pendeteksian Pelanggaran Pada Traffic Light Berbasis Mikrokontroller” penelitian ini dilakukan untuk mendeteksi para pelanggar yang menerobos lampu lalu lintas. Penelitian ini memberikan suatu gambaran dasar bagi penulis terhadap pola-pola dasar dari pemantau pelanggar lalu lintas beberapa tahun kedepan.

Penelitian selanjutnya dilakukan oleh Eric Priyo Tranggono, Anjik Sukmaaji, dan Vicky M Taufik yang berjudul “Rancang Bangun Sistem Informasi Kontrol Kondisi Lalu Lintas Dengan Kamera Pemantau CCTV Berbasis GIS”. Pada penelitian tersebut dilakukan untuk membuat sistem informasi kontrol kondisi lalu lintas dengan pemantau kamera CCTV berbasis GIS yang efektif dan dan bermanfaat, sehingga dapat memudahkan pihak yang berkepentingan untuk melaporkan kondisi suatu jalan serta memudahkan pemantauan dan mengontrol kondisi lalu lintas beserta rute jalan.

### 2.2 Dinas Perhubungan

Dinas Perhubungan kota Surabaya adalah salah satu instansi pemerintah yang mempunyai tugas pokok mengatur segala urusan transportasi dan perhubungan seperti tata letak rambu-rambu lalu lintas, membuat marka jalan, perparkiran, penetapan trayek dan tarif angkutan umum, memeriksa uji kelayakan kendaraan umum. Dinas Perhubungan sendiri mempunyai sistem cerdas untuk mendukung manajemen transportasi dengan pemanfaatan teknologi (informasi, komunikasi, sensor, kontrol dan komputerisasi) untuk membangun sistem informasi dan manajemen transportasi secara otomatis yang biasanya disebut Inteligent Transportation System (ITS).

Dinas Perhubungan kota Surabaya mulai melakukan pemberlakuan CCTV e-tilang sejak bulan September 2017. CCTV e-tilang merupakan penindakan setiap pelanggaran lalu lintas dengan alat bantu kamera CCTV yang terletak pada titik-titik strategis yang sudah di tentukan. Di Kota Surabaya saat ini sedang berjalan 2 titik strategis yang menerapkan CCTV e-tilang, namun di tahun 2018 ini akan ada penambahan 15 unit CCTV. Selain itu, Dinas Perhubungan kota Surabaya juga akan memasang kamera CCTV khusus untuk mengukur kecepatan kendaraan yang melintas di daerah yang rawan mengebut seperti di Kenjeran dan Middle East Ring Road (Peni Widiarti, 2018).

Mekanisme cara kerja CCTV e-tilang ini yaitu pertama, data pelanggar akan terekam di CCTV, data tersebut berisikan nomor polisi pelanggar, jenis pelanggaran, serta tanggal kejadian pelanggar. Kemudian, data yang sudah terekam oleh CCTV akan dikirimkan kepada pihak kepolisian untuk ditindak lanjuti. Jenis pelanggaran yang biasanya terekam oleh CCTV yaitu, melanggar

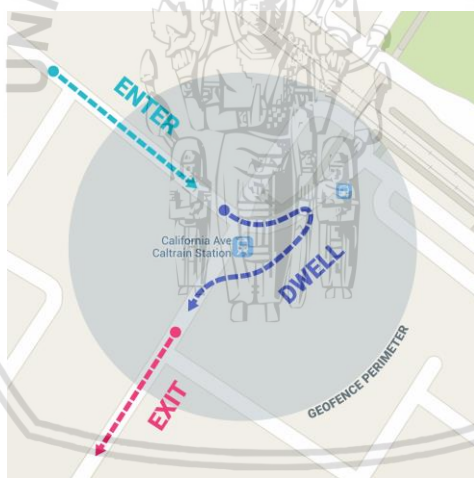
lampu merah, melanggar marka jalan, melawan arah, dan melebihi batas garis saat lampu merah. Tujuan CCTV e-tilang ini adalah untuk mengedukasi masyarakat agar tertib berlalu lintas sehingga mengurangi angka kecelakaan.

### 2.3 Geofencing API

*Geofencing API* adalah sebuah kode program penghubung antara suatu aplikasi dengan suatu fungsi yang telah disediakan oleh Google. Beberapa API lainnya disediakan oleh Google adalah:

1. *Language API*
2. *Earth API*
3. *Javascript API*
4. *Maps API*
5. *Search API*
6. *Visualization API*
7. *YouTube API*

Fungsi *Geofencing API* yaitu memberikan pagar pada suatu wilayah agar pengguna dapat berinteraksi dengan area geografis yang nyata serta dapat mengetahui apakah pengguna berada di dalam atau di luar *geofence* tersebut atau bahkan jika dia keluar atau memasuki area tersebut.



Gambar 2.1 Konsep Dasar Geofence

(Sumber: [www.developers.google.com/location-context/geofencing](http://www.developers.google.com/location-context/geofencing))

### 2.4 Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) yang digunakan untuk pengembangan aplikasi Android (Hafiz Herdi, 2014). Android Studio merupakan IDE resmi untuk Android. Android Studio memiliki fitur editor kode cerdas (*Intelligent Code Editor*) yang memiliki kemampuan penyelesaian kode, optimalisasi, dan analisis kode yang canggih. Selain itu fitur *New Project Wizards* membuat proses memulai proyek baru menjadi jauh lebih mudah bahkan dapat mengimpor contoh



kode Google dari GitHub. Berbagai modul baru digunakan dalam Android Studio ini, salah satunya adalah pengembangan aplikasi multi layar yang memudahkan pengembangan untuk membangun sebuah aplikasi untuk ponsel dan tablet Android, Android Wear, Android TV, Android Auto, dan Android Google Glass. Fitur-fitur Android Studio (Google Inc, 2017) Antara lain sebagai berikut:

1. Dibuat dengan *IntelliJ IDEA Community Edition*, JAVA IDE populer karya JetBrains.
2. Sistem pembuatan *Gradle* yang fleksibel.
3. Dapat membangun berbagai generasi dan varian APK.
4. Dukungan template bertambah untuk Google Services dan aneka tipe perangkat.
5. Editor layout yang lengkap dengan dukungan pengeditan tema.
6. Lint Tools untuk solusi kinerja, kegunaan, kompatibilitas versi, dan masalah lain.
7. ProGuard dan kemampuan app-signing.
8. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah integrasi *Google Cloud Messaging* dan *App-Engine*.

Java dalam ilmu komputer, merupakan bahasa pemrograman berorientasi objek yang diperkenalkan pada tahun 1995 oleh Sun Microsystems Inc., yang saat Java diciptakan, dipimpin oleh James Gosling (Adi Nugroho, 2008). Bahasa pemrograman java tercipta berawal dari sebuah perusahaan Sun Microsystems yang ingin membuat sebuah bahasa pemrograman yang dapat berjalan di semua *device* tanpa harus terikat oleh platform yang digunakan oleh *device* tersebut, terlaksanalah sebuah proyek yang dipelopori oleh Patrick Naughton, James Gosling, Mike Sheridan dan Bill Joy pada tahun 1991, maka terciptalah bahasa pemrograman java yang awalnya bernama "Oak".

Java adalah sebuah teknologi dimana pada teknologi tersebut mencakup java sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman tersendiri, juga mencakup java sebagai platform dimana teknologi ini memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemrograman java (Isak Rickyanto, 2003). Alasan terbesar dalam pembuatan bahasa pemrograman java adalah keinginan akan terbentuknya suatu bahasa pemrograman yang bisa berjalan di berbagai perangkat tanpa harus terikat oleh *platform*, sehingga java ini bersifat *portable* dan *platform independent* ( tidak tergantung mesin atau sistem operasi).

## 2.5 Software Development Life Cycle

*Software Development Life Cycle* (SDLC) merupakan serangkaian aktivitas yang dilakukan untuk menciptakan produk perangkat lunak. Ada beragam proses pengembangan perangkat lunak, namun menurut (Sommerville, 2011) harus memenuhi empat aktivitas yang menjadi dasar perangkat lunak, yaitu :

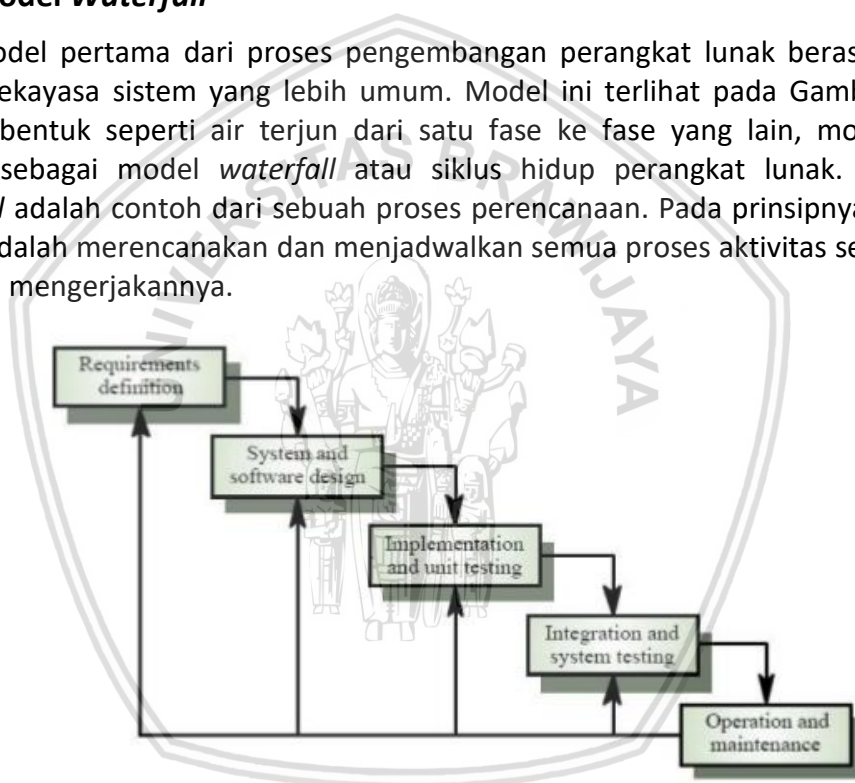


1. *Software specification*, pendefinisian fungsional dan batasan perangkat lunak.
2. *Software design and implementation*, menghasilkan perangkat lunak yang sesuai dengan spesifikasi.
3. *Software validation*, melakukan validasi untuk memastikan bahwa perangkat lunak sesuai dengan kebutuhan pengguna.
4. *Software evolution*, mengembangkan *software* dengan memperbaiki sesuai dengan perubahan kebutuhan pengguna.

Di dalam SDLC terdapat beberapa model, diantaranya adalah model *waterfall* yang akan digunakan dalam penelitian ini.

### 2.5.1 Model *Waterfall*

Model pertama dari proses pengembangan perangkat lunak berasal dari proses rekayasa sistem yang lebih umum. Model ini terlihat pada Gambar 2.1 dengan bentuk seperti air terjun dari satu fase ke fase yang lain, model ini dikenal sebagai model *waterfall* atau siklus hidup perangkat lunak. Model *waterfall* adalah contoh dari sebuah proses perencanaan. Pada prinsipnya, awal proses adalah merencanakan dan menjadwalkan semua proses aktivitas sebelum memulai mengerjakannya.



**Gambar 2.2 Model *Waterfall* (Sumber: Sommerville (2011))**

Berikut tahapan-tahapan utama dalam model *waterfall* yang mencerminkan kegiatan pengembangan yang mendasar :

#### 1. *Requirements Analysis and Definition*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh *software* yang akan dibangun. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan sebagainya.

## 2. *System and Software Design*

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti mengenai domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *User interface* dan sebagainya. Dari dua aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditujukan kepada *user*. Proses *software design* untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk "*blueprint*" *software* sebelum melakukan *coding*. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

## 3. *Implementation and Unit Testing*

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

## 4. *Integration and System Testing*

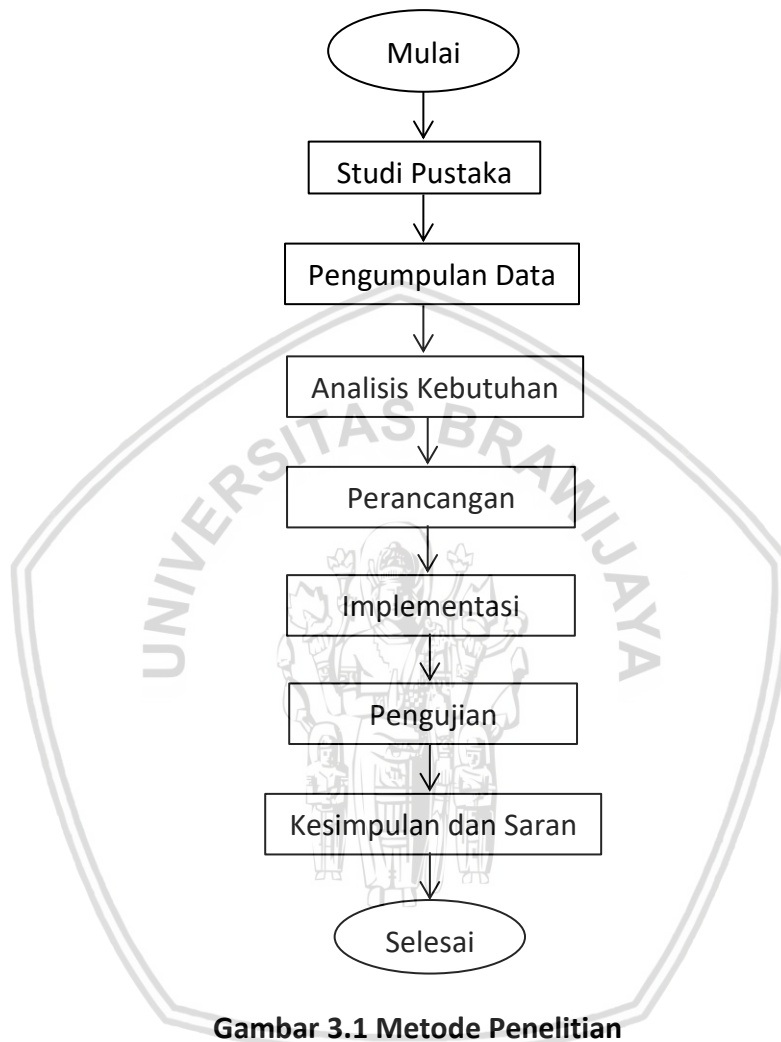
Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. tahap ini merupakan implementasi dari tahap desain yang secara teknis nantinya dikerjakan oleh *programmer*. Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

## 5. *Operation and Maintenance*

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari kesalahan, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya. Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang telah dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin masih terdapat kesalahan kecil yang belum ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika terdapat pergantian sistem operasi atau perangkat lainnya.

## BAB 3 METODOLOGI

Pada bab ini akan diterapkan mengenai penjelasan tentang langkah-langkah dalam pembuatan sebuah perancangan, implementasi serta pengujian sistem. Langkah-langkah tersebut dapat dilihat pada Gambar 3.1.



**Gambar 3.1 Metode Penelitian**

### 3.1 Studi Pustaka

Pada tahap studi pustaka dilakukan pengumpulan informasi atau data dari berbagai macam sumber berupa jurnal, buku, *e-book*, paper penelitian dan internet sebagai rujukan penelitian. Informasi yang dikumpulkan berupa informasi pendukung aplikasi pemantau CCTV e-tilang berbasis android. Hal ini sangat perlu dilakukan agar pengetahuan dasar untuk membangun sebuah sistem terpenuhi dengan baik. Studi literatur yang digunakan meliputi di bawah ini :

1. *Traffic Light E-Tilang*
2. *Software Development Life Cycle (SDLC)*
3. *Unified Modeling Language (UML)*

#### 4. Pengujian Perangkat Lunak

### 3.2 Pengumpulan Data

Tahap pengumpulan data menjelaskan mengenai kegiatan dalam pencarian data yang bertujuan menjawab semua persoalan penelitian. Pada tahap ini, data atau informasi yang dikumpulkan berupa kebutuhan dalam pelanggaran lalu lintas berupa data primer (data yang didapat langsung dari narasumber). Penelitian ini dilaksanakan di Dinas Perhubungan Kota Surabaya dengan melakukan wawancara serta dokumentasi data yang diperoleh berupa pengetahuan pelanggaran lalu lintas dari bulan September 2017 hingga bulan Desember 2017.

### 3.3 Analisis Kebutuhan

Pada tahap analisis kebutuhan dilakukan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (requirement) perangkat lunak dan siapa saja yang terlibat di dalamnya. Analisis juga dilakukan untuk mengetahui kondisi lapangan yang ada sehingga dapat diketahui implementasi perangkat lunak yang akan digunakan. Kebutuhan-kebutuhan yang ditentukan akan dikategorikan menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan-kebutuhan yang ditentukan berasal dari wawancara dengan pihak yang terlibat.

### 3.4 Perancangan

Perancangan perangkat lunak digunakan untuk memenuhi kebutuhan fungsional aplikasi. Tahap Perancangan aplikasi meliputi desain perancangan aplikasi secara umum. Desain perancangan aplikasi secara umum akan menggambarkan bagaimana aplikasi nantinya akan dibuat beserta proses yang terjadi di dalamnya. Serta perancangan antarmuka aplikasi akan menggambarkan desain tampilan aplikasi yang akan dibuat. Dalam perancangan ini metode yang digunakan berbasis objek.

Dalam perancangan ini metode yang digunakan berbasis objek. Terdapat tiga tahapan dalam merancang sistem ini, yaitu perancangan data, perancangan arsitektur, perancangan komponen, dan perancangan *user interface*. Pada perancangan arsitektur diagram yang digunakan adalah *sequence* dan *class* diagram, perancangan komponen yang berupa detail dekomposisi dari arsitektur *class* diagram, dan untuk penghubung antara pengguna dengan sistem yang dibuat melalui perancangan *user interface*.

### 3.5 Implementasi

Implementasi aplikasi dilakukan dengan mengacu kepada perancangan aplikasi. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman JAVA serta aplikasi Android Studio. Model pengembangan sistem

aplikasi ini menggunakan *waterfall* dan menggunakan Firebase sebagai manajemen basis data.

### 3.6 Pengujian

Pengujian perangkat lunak pada penelitian ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang melandainya. Pengujian dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan daftar kebutuhan. Pengujian sistem yang akan dilakukan terdiri dari 2 pengujian, yaitu pengujian validasi dan pengujian performa.

Pengujian validasi dilakukan untuk memastikan bahwa perangkat lunak yang telah di rancang sesuai dengan seluruh kebutuhan konsumen. Pengujian validasi dilakukan dengan menggunakan pengujian *black box*. Sedangkan pengujian performa dilakukan untuk mengetahui keakurasian titik koordinat munculnya notifikasi yang digunakan sebagai pengingat radius lokasi *traffic light*.

### 3.7 Kesimpulan dan Saran

Tahapan-tahapan yang dilakukan dalam penarikan kesimpulan didapatkan dari hasil pembuatan analisis dan pengujian sistem. Hasil kesimpulan diberikan berdasarkan keseluruhan proses pembuatan aplikasi sistem *traffic light* e-tilang untuk menunjukkan apakah masalah penelitian terjawab dan tujuan penelitian tercapai. Sedangkan saran merupakan masukan-masukan yang dibuat untuk memperkecil nilai kesalahan sebagai acuan untuk pengembangan penelitian berikutnya.

## BAB 4 ANALISIS KEBUTUHAN

Dalam analisis kebutuhan yang dilakukan pada bab ini menjelaskan gambaran umum sistem yang akan dibangun, aktor yang akan menggunakan aplikasi dan kebutuhan dalam membangun sistem.

### 4.1 Gambaran Umum Sistem

Sistem *traffic light* e-tilang merupakan sebuah perangkat lunak berbasis Android, dimana pengguna dapat secara langsung mengaksesnya pada *device* Android. Kegunaan sistem informasi ini untuk mempermudah serta membantu masyarakat dalam mengetahui jenis pelanggaran, waktu pelanggaran serta lokasi pelanggaran yang telah dilanggar. Khususnya untuk masyarakat umum kota Surabaya. Masyarakat juga dapat melakukan panggilan darurat seperti kepolisian, ambulans, dan pemadam kebakaran yang tersedia di dalam aplikasi.

### 4.2 Analisis Data

Analisis data dilakukan dengan tujuan untuk menentukan struktur penyimpanan data yang akan dibuat. Data yang disimpan adalah data yang diharapkan dapat berguna bagi pengguna aplikasi. Dalam mengetahui data yang akan digunakan dilakukan dengan wawancara pada pihak Dinas Perhubungan Kota Surabaya. Dari hasil wawancara tersebut mendapatkan dokumentasi data informasi tilang seperti nomor polisi kendaraan, waktu, lokasi, dan jenis pelanggaran.

Pengumpulan data selanjutnya dilakukan dengan pengisian kuesioner pada masyarakat dengan jumlah responden 30 orang. Tujuan dari kuesioner adalah untuk mengetahui daftar kebutuhan pengguna. Dengan mengetahui daftar kebutuhan pengguna, maka akan di dapatkan daftar kebutuhan untuk menunjang kebutuhan fungsional dari sistem *traffic light* e-tilang. Jenis kebutuhan dan deskripsi ditunjukkan pada Tabel 4.1.

**Tabel 4.1 Analisis Data Kebutuhan**

Jenis Data Kebutuhan	Deskripsi
Data nomor polisi kendaraan	Digunakan untuk login, saat pertama kali pengguna menjalankan aplikasi.
Data waktu pelanggaran	Digunakan untuk mengetahui waktu pengguna ketika melanggar lalu lintas.
Data lokasi pelanggaran	Digunakan untuk mengetahui lokasi pengguna ketika melanggar lalu lintas.
Data jenis pelanggaran	Digunakan untuk mengetahui jenis pelanggaran pengguna ketika melanggar lalu lintas.



### 4.3 Identifikasi Aktor

Aktor merupakan seseorang ataupun sistem yang dapat berinteraksi dengan sistem. Adapun aktor dalam aplikasi ini ditunjukkan pada Tabel 4.2.

**Tabel 4.2 Identifikasi Aktor**

Aktor	Deskripsi
Pengguna	Merupakan seseorang yang mempunyai hak akses untuk menggunakan semua fitur dari aplikasi yang telah dibuat.

### 4.4 Kebutuhan Fungsional Sistem

Kebutuhan fungsional sistem merupakan layanan dari sebuah sistem yang harus tersedia, bagaimana sistem seharusnya merespon pada masukan tertentu dan bagaimana sistem harus melakukan apa pada situasi tertentu (Kurniawan, 2016). Kebutuhan fungsional harus mencakup apa saja yang harus diselesaikan oleh sistem untuk dapat menyelesaikan masalah yang telah ditentukan sebelumnya. Kebutuhan fungsional juga harus diberikan suatu identitas untuk mempermudah proses identifikasi kebutuhan juga untuk menjaga konsistensi yang terkait dengan kebutuhan suatu sistem, mulai dari proses perancangan hingga pengujian sistem selesai dilakukan. Adapun kebutuhan fungsional sistem serta spesifikasinya ditunjukkan pada Tabel 4.3. Setiap Kebutuhan akan diberikan kode SPTL-F-X untuk kebutuhan fungsional dan SPTL-NF-X untuk kebutuhan non fungsional. SPTL merupakan Sistem Pemantau *Traffic Light*, F untuk kebutuhan fungsional, NF untuk kebutuhan non fungsional dan X menunjukkan nomor dari definisi kebutuhan utama.

**Tabel 4.3 Kebutuhan Fungsional Sistem**

No.	Kode	Nama Fungsi	Deskripsi Kebutuhan
1	SPTL-F-001	Login	Aplikasi dapat melakukan autentifikasi pengguna agar dapat masuk ke dalam sistem dengan melakukan pengisian nomor polisi kendaraan aktor dan pemindaian kode QR.
2	SPTL-F-002	Melihat informasi pelanggaran dan estimasi denda	Aplikasi dapat menampilkan informasi jenis pelanggaran seperti jenis pelanggaran, waktu pelanggaran, denda setiap pelanggaran dan lokasi pelanggaran pengguna.

**Tabel 4.4 Kebutuhan Fungsional Sistem (Lanjutan)**

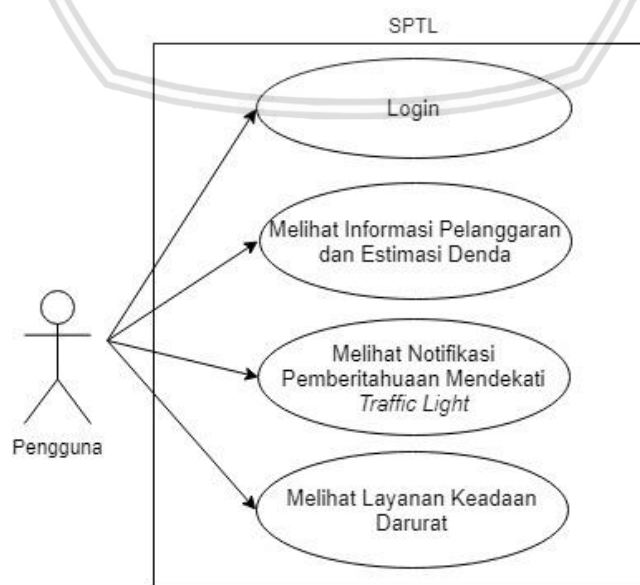
No.	Kode	Nama Fungsi	Deskripsi Kebutuhan
3	SPTL-F-003	Melihat notifikasi pemberitahuan mendekati <i>traffic light</i>	Aplikasi dapat menampilkan notifikasi peringatan di suatu titik terjadinya banyak pelanggaran lalu lintas yang muncul hanya satu kali.
4	SPTL-F-004	Melihat layanan keadaan darurat	Aplikasi akan menampilkan beberapa nomor yang akan digunakan aktor untuk menghubungi pihak-pihak yang berhubungan dengan keadaan darurat yang sedang dialami.

**Tabel 4.5 Kebutuhan Non Fungsional Sistem**

No.	Kode	Nama Fungsi	Deskripsi Kebutuhan
1	SPTL-NF-001	<i>Usability</i>	Kemudahan dalam menggunakan aplikasi yang digunakan oleh pengguna.

#### 4.5 Diagram Use Case

Diagram *use case* berisi sejumlah aksi yang dilakukan oleh aktor kepada sistem untuk mencapai tujuan tertentu. Aktor dalam diagram *use case* bisa berupa manusia atau sistem eksternal yang berinteraksi dengan sistem yang dimaksud. Diagram *use case* ditunjukkan dalam Gambar 4.1.



**Gambar 4.1 Diagram Use Case Sistem**

#### 4.6 Skenario Use Case

Skenario *use case* merupakan penjabaran lebih detail dari tiap *use case* yang digambarkan pada diagram *use case*. Pada skenario *use case* terdapat beberapa bagian, diantaranya nama *use case*, kode kebutuhan, aktor yang berinteraksi, tujuan *use case*, *pre condition*, *main flow*, *alternative flow* & *post condition*. Pada bagian *pre condition*, berisi berbagai kondisi yang harus dipenuhi oleh sistem sebelum menjalankan *main flow* pada *use case*. Sedangkan *main flow* sendiri berisi alur-alur pengerjaan *use case* hingga selesai mendapatkan hasil yang diharapkan pada *post condition*, *alternative flow* sendiri berisi alur pengerjaan ketika dalam pengerjaan *main flow* terjadi skenario yang tidak dideklarasikan pada *main flow*, sedangkan *post condition* berisi kondisi akhir setelah aktor menjalankan *main flow*. Diperoleh 4 *use case scenario* yaitu login, melihat informasi pelanggaran, melihat kalkulasi estimasi denda, dan melihat layanan keadaan darurat. Skenario *use case* akan ditunjukkan oleh Tabel 4.3 hingga Tabel 4.6.

**Tabel 4.6 Skenario Use Case Login**

Nama Use Case	Login
Kode Kebutuhan Terkait	SPTL-F-001
Aktor	Pengguna
Pre Condition	Pengguna telah membuka halaman informasi pelanggaran.
Main flow	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan nomor polisi kendaraan.</li> <li>2. Pengguna melakukan pemindaian kode QR.</li> <li>3. Sistem menampilkan informasi pelanggaran.</li> </ol>
Alternatif Flow	<ol style="list-style-type: none"> <li>1. Sistem menampilkan pesan "Data tidak ditemukan."</li> </ol>
Post Condition	Sistem menampilkan informasi pelanggaran sesuai dengan masukan pengguna.

**Tabel 4.7 Skenario Use Case Melihat Informasi Pelanggaran dan Estimasi Denda**

Nama Use Case	Melihat informasi pelanggaran dan estimasi denda
Kode Kebutuhan Terkait	SPTL-F-002
Aktor	Pengguna
Pre Condition	Pengguna telah membuka halaman utama.

**Tabel 4.7 Skenario *Use Case* Melihat Informasi Pelanggaran dan Estimasi Denda (Lanjutan)**

<i>Main flow</i>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol Pencarian Data.</li> <li>2. Sistem menampilkan informasi pelanggaran dan estimasi denda.</li> </ol>
<i>Alternatif Flow</i>	-
<i>Post Condition</i>	Sistem menampilkan informasi pelanggaran dan estimasi denda sesuai dengan masukan pengguna.

**Tabel 4.8 Skenario *Use Case* Melihat Notifikasi Pemberitahuan Mendekati *Traffic Light***

Nama <i>Use Case</i>	Melihat pemberitahuan mendekati <i>traffic light</i>
Kode Kebutuhan Terkait	SPTL-F-003
Aktor	Pengguna
<i>Pre Condition</i>	Pengguna berada di posisi radius 1km dari jarak <i>traffic light</i> dan GPS telah aktif.
<i>Main flow</i>	<ol style="list-style-type: none"> <li>1. Sistem mengetahui posisi pengguna berada di dalam radius 1km dari <i>traffic light</i>.</li> <li>2. Sistem menampilkan pemberitahuan mendekati <i>traffic light</i>.</li> </ol>
<i>Alternatif Flow</i>	-
<i>Post Condition</i>	Sistem menampilkan pemberitahuan mendekati <i>traffic light</i> .

**Tabel 4.9 Skenario *Use Case* Melihat Layanan Keadaan Darurat**

Nama <i>Use Case</i>	Melihat layanan keadaan darurat
Kode Kebutuhan Terkait	SPTL-F-004
Aktor	Pengguna
<i>Pre Condition</i>	Pengguna telah membuka halaman utama.
<i>Main flow</i>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol Layanan Darurat.</li> <li>2. Sistem menampilkan beberapa nomor telepon layanan darurat.</li> </ol>
<i>Alternatif Flow</i>	-

**Tabel 4.8 Skenario *Use Case* Melihat Layanan Keadaan Darurat (Lanjutan)**

<i>Post Condition</i>	Sistem menampilkan informasi nomor telepon layanan darurat
-----------------------	--



## BAB 5 PERANCANGAN DAN IMPLEMENTASI

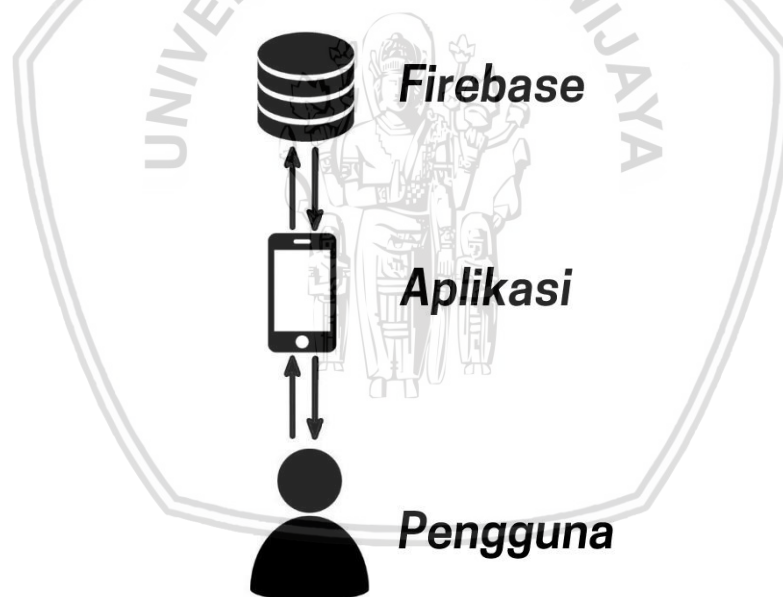
Sebelum mengimplementasikan sistem, dilakukan perancangan sistem yang digunakan sebagai dasar untuk mengimplementasikan sistem, dalam tahap ini perancangan yang akan dilakukan adalah perancangan arsitektur sistem, perancangan komponen, perancangan data dan perancangan antar muka.

### 5.1 Perancangan

Setelah proses analisis kebutuhan selesai dilakukan, tahap berikutnya adalah perancangan. Perancangan dilakukan berdasarkan hasil dari analisis kebutuhan yang telah dilakukan. Dalam perancangan sistem terdiri dari perancangan arsitektur sistem, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, perancangan *user interface*, dan perancangan algoritme.

#### 5.1.1 Perancangan Arsitektur Sistem

Perancangan arsitektur untuk sistem *traffic light* e-tilang dapat dilihat pada Gambar 5.1.



Gambar 5.1 Perancangan Arsitektur Sistem

Berikut penjelasan mengenai masing-masing entitas pada Gambar 5.1.

a. Pengguna

Pengguna merupakan entitas yang berinteraksi langsung dengan aplikasi. Pengguna dapat memanfaatkan semua fitur yang terdapat di dalam aplikasi.



b. Aplikasi

Aplikasi merupakan entitas yang digunakan untuk menampilkan *user interface* secara langsung pada pengguna. Aplikasi juga memiliki fungsi perantara antara data dan pengguna yang ditampilkan melalui antar muka aplikasi.

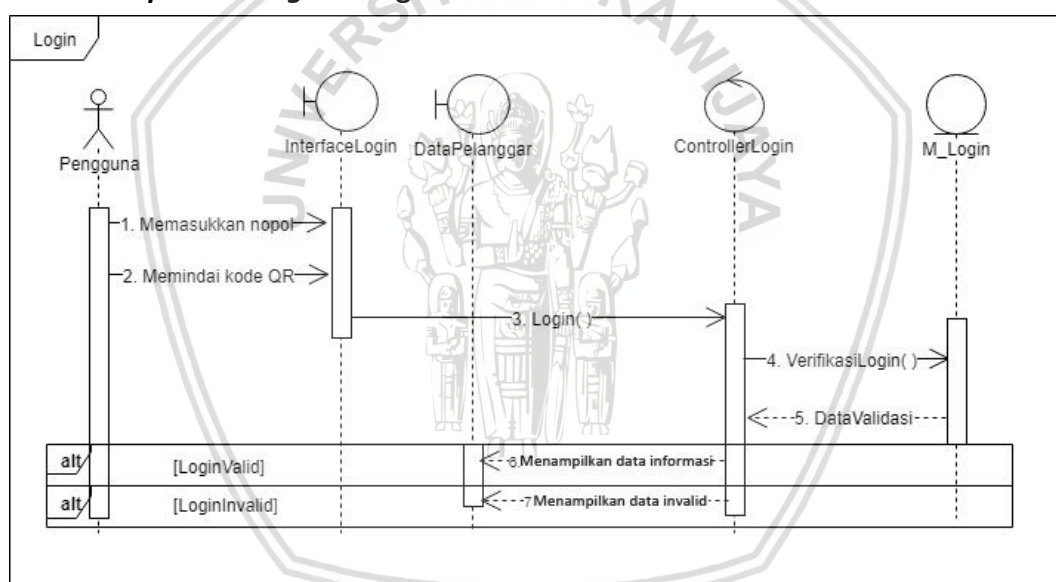
c. Firebase

Database digunakan untuk menyimpan data yang akan ditampilkan mauun digunakandalam aplikasi.

### 5.1.2 Perancangan *Sequence Diagram*

Pada pemodelan *sequence diagram* dijelaskan bahwa urutan proses yang terjadi untuk mencapai suatu kebutuhan sistem. Semua objek pada *sequence diagram* merupakan hasil identifikasi dari spesifikasi kebutuhan dan *use case scenario* yang ada pada tahap analisis kebutuhan sebelumnya. Setiap *sequence diagram* menggambarkan setiap *use case* yang ada.

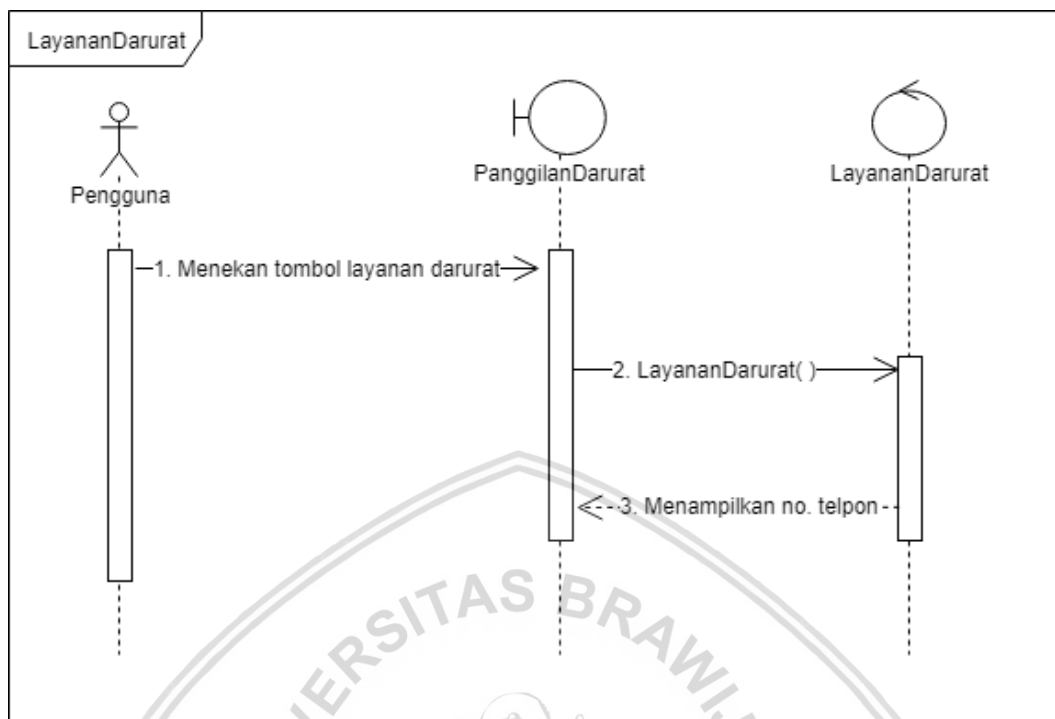
#### 5.1.2.1 *Sequence Diagram Login*



**Gambar 5.2 *Sequence Diagram Login***

*Sequence diagram* login dapat dilihat pada Gambar 5.2. Dalam *sequence diagram* login dijelaskan bahwa proses login dengan aktor pengguna. Untuk dapat melakukan proses login, pengguna memasukkan nopol kendaraan dan memindai kode QR, kemudian sistem akan mengirim data login yang diisikan ke *ControllerLogin* untuk kemudian di cek ke model. Jika data benar, maka akan menampilkan halaman utama data pelanggaran. Jika salah, maka akan menampilkan pesan error.

### 5.1.2.2 Sequence Diagram Layanan Keadaan Darurat

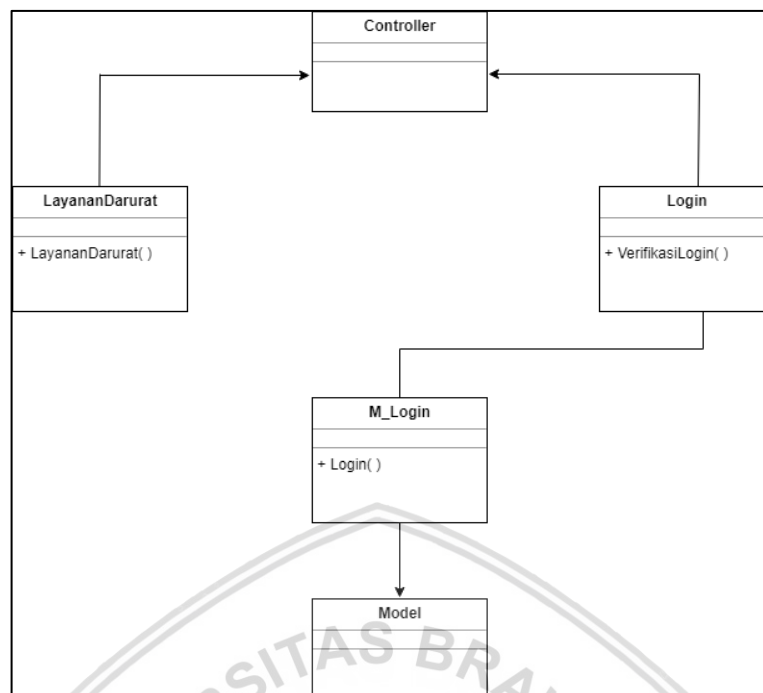


**Gambar 5.3 Sequence Diagram Layanan Darurat**

*Sequence diagram* layanan darurat dapat dilihat pada Gambar 5.3. Dalam *sequence diagram* layanan darurat dijelaskan bahwa proses layanan darurat dengan aktor pengguna. Untuk dapat melakukan proses layanan darurat, pengguna menekan tombol layanan darurat, kemudian *Boundary* PanggilanDarurat akan memanggil fungsi LayananDarurat( ) pada *Controller* LayananDarurat. Setelah itu, sistem akan menampilkan nomor telepon kepada pengguna.

### 5.1.3 Perancangan Class Diagram

Mengacu pada *sequence diagram* secara implementasi, pada Gambar 5.4 menjelaskan *class diagram* dari sistem *traffic light* e-tilang. *Class diagram* tersebut menjelaskan fungsi-fungsi yang ada pada setiap *class* dan tipe data yang dimiliki oleh fungsi tersebut. Pada Gambar 5.4 terdapat beberapa *class diagram* yang mempunyai fungsi masing-masing. Yang bertindak sebagai model adalah database, sedangkan *Controller* merupakan pengontrol jalannya sistem.



Gambar 5.4 Perancangan Class Diagram

#### 5.1.4 Perancangan Basis Data

Dalam sistem *traffic light* e-tilang diperlukan *database* untuk menyimpan data informasi pelanggaran. Maka diperlukan sebuah perancangan basis data yang digunakan untuk mempermudah pemahaman data yang akan dimasukkan ke dalam sistem. Dalam perancangan basis data, *database* yang digunakan adalah Firebase.

Perancangan basis data pada sistem ini menggunakan JSON Schema. JSON Schema menyediakan kontrak untuk data JSON yang diperlukan untuk aplikasi tertentu dan cara berinteraksi dengannya. JSON Schema digunakan untuk menentukan validasi, dokumentasi, navigasi hyperlink, dan kontrol interaksi data JSON. Gambar 5.5 menunjukkan perancangan sistem *traffic light* e-tilang.



Gambar 5.5 Perancangan Class Diagram

Terdapat tabel Informasi yang merepresentasikan informasi pelanggaran pengguna. Tabel informasi digunakan untuk menyimpan data nomor polisi kendaraan, waktu, lokasi, jenis pelanggaran, dan denda. Dalam tabel ini nopol kendaraan digunakan sebagai primary key. Perancangan tabel login dapat dilihat pada Tabel 5.1

**Tabel 5.1 Perancangan Basis Data Tabel Informasi**

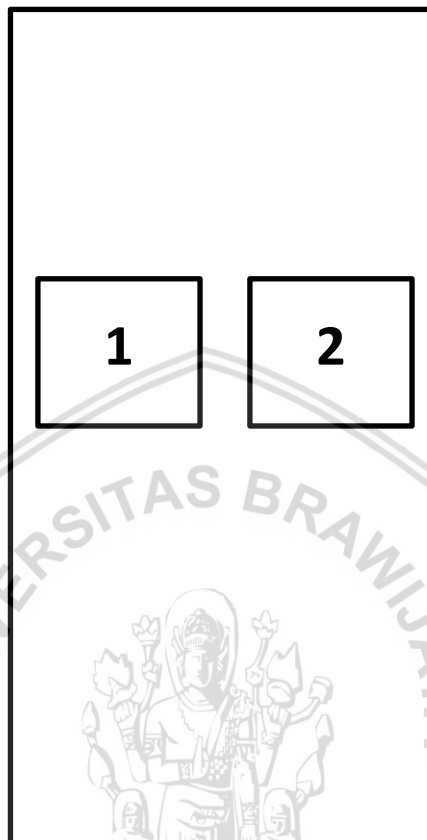
No	Nama Field	Tipe Data	Panjang / Nilai	Keterangan
1	Nopol Kendaraan (PK)	<i>Integer</i>	8	Merupakan <i>primary key</i> dari tabel login dan berisi.
2	Waktu	<i>Varchar</i>	5	Berisi waktu dari pelanggaran yang terekam oleh CCTV.
3	Lokasi	<i>Varchar</i>	50	Berisi lokasi dari pelanggaran yang terekam oleh CCTV.
4	Jenis Pelanggaran	<i>Varchar</i>	100	Berisi jenis pelanggaran yang terekam oleh CCTV.
5	Denda	<i>Varchar</i>	15	Berisi denda dari masing-masing jenis pelanggaran.

#### 5.1.5 Perancangan Antarmuka

Perancangan antarmuka merupakan kerangka pembuatan sebuah aplikasi. Tujuan dari perancangan antarmuka untuk membuat sebuah tampilan dari aplikasi yang sederhana agar memudahkan pengguna dalam mengoperasikan aplikasi. Pada bagian ini akan dijelaskan antarmuka sistem *traffic light* e-tilang. Perancangan antarmuka terdiri dari rancangan menu utama, *input* nomor polisi, pindai kode, informasi pelanggaran, dan panggilan darurat.

#### 5.1.5.1 Perancangan Antarmuka Menu Utama

Perancangan antarmuka menu utama pada Gambar 5.6 dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.2.



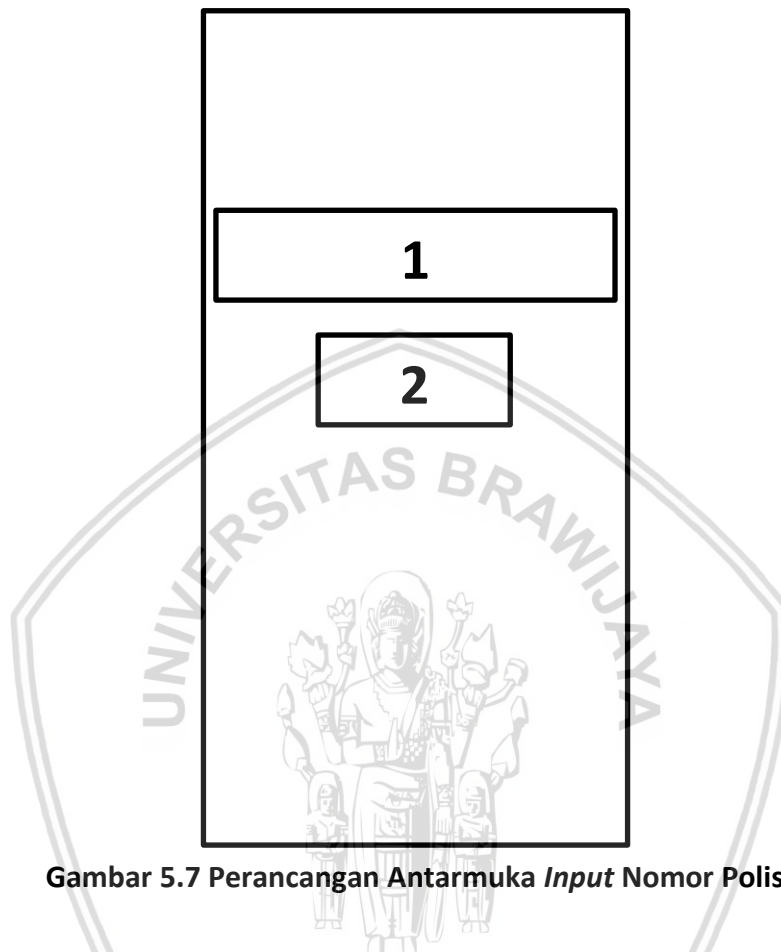
Gambar 5.6 Perancangan Antarmuka Menu Utama

Tabel 5.2 Penjelasan Antarmuka Menu Utama

No.	Nama Objek	Tipe	Keterangan
1	Panggilan Darurat	<i>Button</i>	Tombol tersebut akan mengarahkan pengguna menuju halaman panggilan darurat.
2	Informasi Pelanggaran	<i>Button</i>	Tombol tersebut akan mengarahkan pengguna menuju halaman informasi pelanggaran.

#### 5.1.5.2 Perancangan Antarmuka *Input* Nomor Polisi

Perancangan antarmuka input nomor polisi pada Gambar 5.7 dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.3.



Gambar 5.7 Perancangan Antarmuka *Input* Nomor Polisi

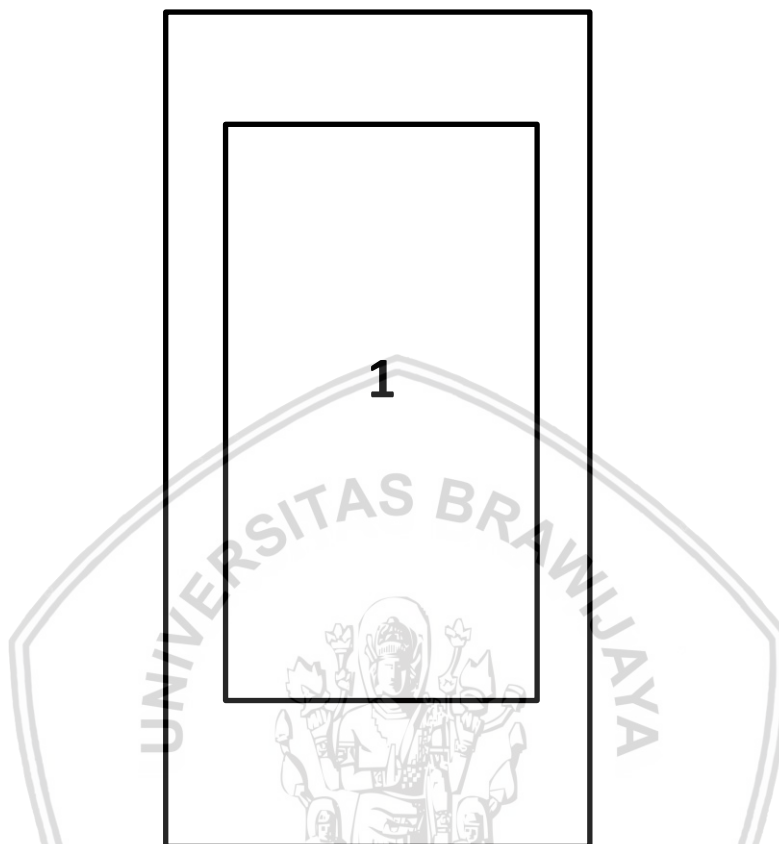
Tabel 5.3 Penjelasan Antarmuka *Input* Nomor Polisi

No.	Nama Objek	Tipe	Keterangan
1	Nomor polisi kendaraan	<i>Form field</i>	Form untuk mengisi nomor polisi kendaraan.
2	Pindai kode	<i>Button</i>	Tombol tersebut akan mengarahkan pengguna menuju halaman pindai kode QR.



### 5.1.5.3 Perancangan Antarmuka Pindai Kode

Perancangan antarmuka pindai kode pada Gambar 5.8 dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.4.



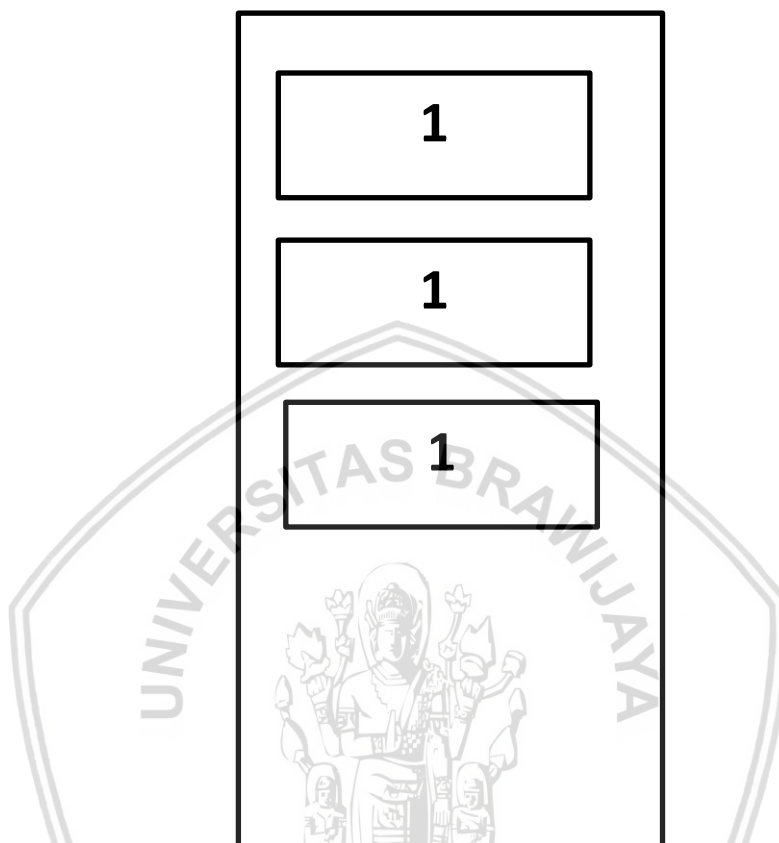
**Gambar 5.8 Perancangan Antarmuka Pindai Kode**

**Tabel 5.4 Penjelasan Antarmuka Pindai Kode**

No.	Nama Objek	Tipe	Keterangan
1	Kode QR	<i>Scanner camera</i>	Scanner tersebut digunakan untuk memindai kode QR melalui kamera ponsel

#### 5.1.5.4 Perancangan Antarmuka Informasi Pelanggaran

Perancangan antarmuka informasi pelanggaran pada Gambar 5.9 dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.5.



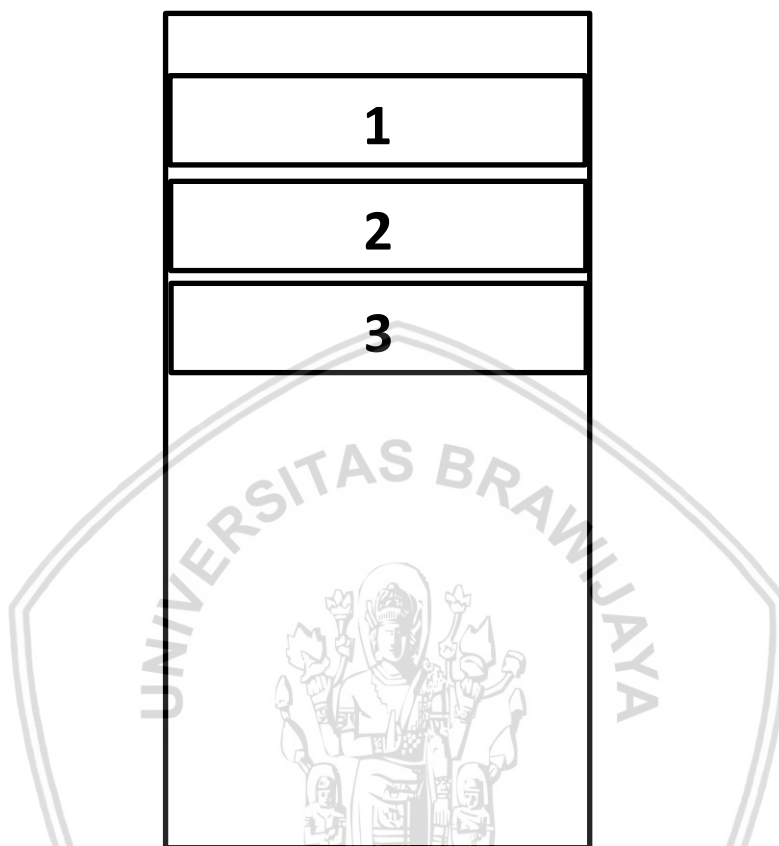
**Gambar 5.9 Perancangan Antarmuka Informasi Pelanggaran**

**Tabel 5.5 Penjelasan Antarmuka Informasi Pelanggaran**

No.	Nama Objek	Tipe	Keterangan
1	Informasi pelanggaran	Tabel	Tabel tersebut berisi informasi pelanggaran seperti nomor polisi kendaraan, waktu pelanggaran, lokasi pelanggaran, jenis pelanggaran, dan denda.

#### 5.1.5.5 Perancangan Antarmuka Panggilan Darurat

Perancangan antarmuka panggilan darurat pada Gambar 5.10 dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.6.



Gambar 5.10 Perancangan Antarmuka Panggilan Darurat

Tabel 5.6 Penjelasan Antarmuka Panggilan Darurat

No.	Nama Objek	Tipe	Keterangan
1	Pemadam kebakaran	<i>Button</i>	Tombol tersebut akan mengarahkan pengguna menuju nomor telepon pemadam kebakaran.
2	Ambulans	<i>Button</i>	Tombol tersebut akan mengarahkan pengguna menuju nomor telepon pemadam kebakaran.
3	Polisi	<i>Button</i>	Tombol tersebut akan mengarahkan pengguna menuju nomor telepon pemadam kebakaran.

### 5.1.6 Perancangan Algoritme

Perancangan algoritme menggambarkan rincian sub-sistem dari setiap komponen perangkat lunak. Untuk mencapai hal ini, perancangan algoritme harus mendefinisikan struktur data untuk semua objek data lokal dan rincian algoritme untuk proses yang terjadi di dalam komponen. Dalam perancangan komponen ini dipilih satu Algoritme *method* utama dari setiap *class* pada *controller*.

#### 5.1.6.1 Perancangan Algoritme Login

Perancangan algoritme login dapat dilihat pada Tabel 5.8

**Tabel 5.7 Perancangan Algoritme Login**

```

Start
Menekan tombol Informasi Pelanggaran
    Menampilkan halaman input data berupa Form input
Input Nopol Kendaraan
Memindai kode QR
    Jika Nopol Kendaraan == null
        Sistem menampilkan pesan "Field is empty"
    Selain itu
        Membuka activity Scan
    Jika, kode QR != nopol
        Sistem menampilkan notifikasi "Data tidak di temukan"
    Selain itu
        Sistem menampilkan halaman informasi pelanggaran
End
  
```

#### 5.1.6.2 Perancangan Algoritme Notifikasi

Perancangan algoritme notifikasi dapat dilihat pada Tabel 5.9

**Tabel 5.8 Perancangan Algoritme Notifikasi**

```

Start
Inisialisasi lokasi
for i =0 to locationList.lenght do
    if locationList[i] closer to location do
        send notifikasi
    end if
next i
End
  
```

## 5.2 Implementasi

Pada bab implementasi akan dijelaskan mengenai implementasi yang dilakukan dalam pembuatan sistem. Sebelum implementasi dijelaskan spesifikasi yang digunakan dalam implementasi sistem. Spesifikasi tersebut berupa spesifikasi sistem yang digunakan, serta batasan dalam implementasi.

### 5.2.1 Spesifikasi Sistem

Dalam mengimplementasikan hasil dari analisis kebutuhan, perancangan, dan hasil dari evaluasi pengguna dibutuhkan perangkat keras serta perangkat lunak. Perangkat keras dan perangkat lunak perlu dispesifikasikan agar dapat mengetahui standar yang diperlukan untuk pembuatan aplikasi selama dalam penelitian.

#### 5.2.1.1 Spesifikasi Perangkat Keras

Dalam implementasi aplikasi, dibutuhkan perangkat keras berupa komputer dan *smartphone*. Komputer digunakan untuk melakukan implementasi atau pembuatan aplikasi mulai dari implementasi antarmuka hingga implementasi algoritma. Sementara *smartphone* digunakan untuk pengujian atau perangkat penguji aplikasi. Spesifikasi perangkat keras komputer yang digunakan dalam implementasi dapat dilihat pada Tabel 5.10.

**Tabel 5.9 Spesifikasi Perangkat Keras Komputer**

No	Nama Komponen	Spesifikasi
1	Model	Asus X455LF
2	<i>Processor</i>	Intel(R) Core i3-4005U, CPU 1.70GHz
3	<i>Memory</i>	6GB RAM, 500GB HDD
4	<i>Display</i>	Nvidia Geforce 930M

Spesifikasi perangkat keras *smartphone* yang digunakan dapat dilihat pada Tabel 5.11.

**Tabel 5.10 Spesifikasi Perangkat Keras *Smartphone***

No	Nama Komponen	Spesifikasi
1	Model	Samsung J7+
2	<i>Processor</i>	Octa-core 2.39GHz, 1.69 GHz
3	<i>Memory</i>	4GB RAM, 32GB internal memory
4	<i>Display</i>	5,5 inch

### 5.2.1.2 Spesifikasi Perangkat Lunak

Dalam implementasi aplikasi, dibutuhkan perangkat lunak yang terdapat dalam komputer dan *smartphone*. Spesifikasi perangkat lunak komputer yang digunakan dalam implementasi dapat dilihat pada Tabel 5.12.

**Tabel 5.11 Spesifikasi Perangkat Lunak Komputer**

No	Nama Komponen	Spesifikasi
1	Sistem Operasi	Windows 10 Pro
2	Bahasa Pemrograman	Java
3	<i>Engine</i>	Android Studio
4	<i>Database server</i>	Firebase

Sedangkan untuk spesifikasi perangkat lunak *Smartphone* dapat dilihat pada Tabel 5.13.

**Tabel 5.12 Spesifikasi Perangkat Lunak Smartphone**

No	Nama Komponen	Spesifikasi
1	Sistem Operasi	Android 8.0.0

### 5.2.2 Implementasi Algoritma

#### 1. Algoritma Login

Fungsi yang digunakan untuk login adalah method `onClick`. Pada fungsi ini format masukan pengguna akan divalidasi, jika terdapat masukan yang salah maka sistem menampilkan pesan yang salah. Pada method `handleResult` digunakan untuk mendapatkan hasil dari pemindaian kode QR sekaligus mencocokkan kode QR dengan input nomor polisi kendaraan. Method `onClick` dan method `handleResult` termuat dalam Tabel 5.14 sebagai berikut.

**Tabel 5.13 Implementasi Algoritma Login**

1	<code>public void onClick(View v) {</code>
2	<code>    switch (v.getId()){</code>
3	<code>        case R.id.imageButton2:</code>
4	<code>            String mytext = inputnopol.getText().toString();</code>
5	<code>            if (TextUtils.isEmpty(mytext)) {</code>
6	<code>                Toast.makeText(Input_nopol.this, "Fields is</code>
7	<code>empty", Toast.LENGTH_LONG).show();</code>
8	<code>            } else {</code>
9	<code>                Intent intent = new Intent(Input_nopol.this,</code>
10	<code>Scan_qr.class);</code>
11	<code>                intent.putExtra("Nopol", mytext);</code>
12	<code>                startActivity(intent);</code>
13	<code>            } break;</code>
14	<code>        }</code>
15	<code>    }</code>
16	<code>public void handleResult(Result result) {</code>
17	<code>    final String myResult = result.getText();</code>



**Tabel 5.14 Implementasi Algoritma Login (Lanjutan)**

14	final String tempNopol = nopol;
15	this.nopol = null;
16	if (myResult.equalsIgnoreCase(tempNopol)){
17	Intent intent = new Intent(Scan_qr.this,
18	Info_pelanggaran.class);
19	intent.putExtra("Nopol", myResult);
20	startActivity(intent);
21	}else{
22	Toast.makeText(Scan_qr.this, "Data tidak cocok",
	Toast.LENGTH_LONG).show();
	finish();
	}

## 2. Algoritma Notifikasi

Fungsi yang digunakan untuk notifikasi adalah method update, yang digunakan untuk mengecek lokasi dan memberikan peringatan bahwa *location* yang ada di *smartphone* harus terhubung terlebih dahulu dan mengarahkan ke method addProximityAlert yang berfungsi untuk mengirimkan kode lokasi longitude dan latitude ke *activity* AlertOnProximityReceiver. Di dalam *class* AlertOnProximityReceiver terdapat algoritma method onReceive yang berfungsi untuk menginisialisasi nama lokasi dan menyimpan fungsi ketika pengguna mendekati lokasi yang telah disimpan oleh sistem dan method sendNotification berfungsi untuk memanggil notifikasi yang telah tersimpan di dalam method onReceive.

**Tabel 5.14 Implementasi Algoritma Notifikasi**

1	public void update(){
2	if (ActivityCompat.checkSelfPermission(this,
3	android.Manifest.permission.ACCESS_FINE_LOCATION) !=
4	PackageManager.PERMISSION_GRANTED &&
5	ActivityCompat.checkSelfPermission(this,
6	android.Manifest.permission.ACCESS_COARSE_LOCATION)
7	!= PackageManager.PERMISSION_GRANTED) {
8	Toast.makeText(MainActivity.this, "First enable
9	UPDATE_LOCATION ACCESS in settings.", Toast.LENGTH_LONG).show();
10	return;
11	}
12	locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
13	0, 0, this);
14	for (Map.Entry<String,Double[]> entry :
15	locationList.entrySet()) {
16	String key = entry.getKey();
17	Double[] value = entry.getValue();
18	Double lati = value[0];
19	Double longi = value[1];
20	addProximityAlert(lati, longi, 100, key);
21	}
22	private void addProximityAlert(double latitude, double
23	longitude, int radius, String poiName) {
24	
25	String newIntentName = PROX_ALERT_INTENT + "." +

**Tabel 5.15 Implementasi Algoritma Notifikasi (Lanjutan)**

	requestCode;
16	Intent intent = new Intent(newIntentName);
17	intent.putExtra("name", poiName);
18	intent.putExtra("id", requestCode);
19	PendingIntent proximityIntent =
20	PendingIntent.getBroadcast(this, requestCode, intent,
21	PendingIntent.FLAG_CANCEL_CURRENT);
22	if (ActivityCompat.checkSelfPermission(this,
23	android.Manifest.permission.ACCESS_FINE_LOCATION) !=
24	PackageManager.PERMISSION_GRANTED &&
25	ActivityCompat.checkSelfPermission(this,
26	android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
27	PackageManager.PERMISSION_GRANTED) {
28	return;
29	}
30	locationManager.addProximityAlert(latitude, longitude,
31	(float) radius, -1, proximityIntent);
32	IntentFilter filter = new IntentFilter(newIntentName);
33	registerReceiver(new AlertOnProximityReceiver(new
34	Messenger(new MainHandler(this))), filter, null, backgroundHandler);
35	requestCode++;
36	}
37	public void onReceive(final Context context, final Intent intent) {
38	this.notificationManager
39	= (NotificationManager)
40	context.getSystemService(Context.NOTIFICATION_SERVICE);
41	String key = LocationManager.KEY_PROXIMITY_ENTERING;
42	Boolean getting_closer = intent.getBooleanExtra(key, false);
43	final String locationName = intent.getExtras().getString
44	("name");
45	if (getting_closer) {
46	sendNotification(context, locationName, myLat, myLong);
47	try {
48	messenger.send(location);
49	messenger.send(status);
50	} catch (RemoteException e) {
51	e.printStackTrace();
52	}
53	}
54	public void sendNotification(Context mContext, String msg,
55	String myLat, String myLong) {
56	Intent notificationIntent = new Intent(mContext,
57	MainActivity.class);
58	notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP
59	Intent.FLAG_ACTIVITY_SINGLE_TOP);
60	PendingIntent intent = PendingIntent.getActivity(mContext,
61	0, notificationIntent, 0);
62	this.myNotification =
63	new NotificationCompat.Builder(mContext)
64	.setContentTitle("Judul Notifikasi")
65	.setContentText(msg + " " + myLat + "," + myLong)
66	.setSmallIcon(R.drawable.ic_traffic_black_24dp)
67	.setContentIntent(intent)
68	.setChannelId("Ch01")
69	.build();
70	if (android.os.Build.VERSION.SDK_INT >=

**Tabel 5.15 Implementasi Algoritma Notifikasi (Lanjutan)**

53	android.os.Build.VERSION_CODES.O) {
54	NotificationChannel channel = new NotificationChannel(
55	"Ch01", "Channel 01",
56	NotificationManager.IMPORTANCE_HIGH);
57	this.notificationManager.createNotificationChannel(channel);
58	}
	this.notificationManager.notify(
	99,
	this.myNotification);
	}

### 5.2.3 Implementasi Antarmuka

#### 1. Implementasi Antarmuka Menu Utama

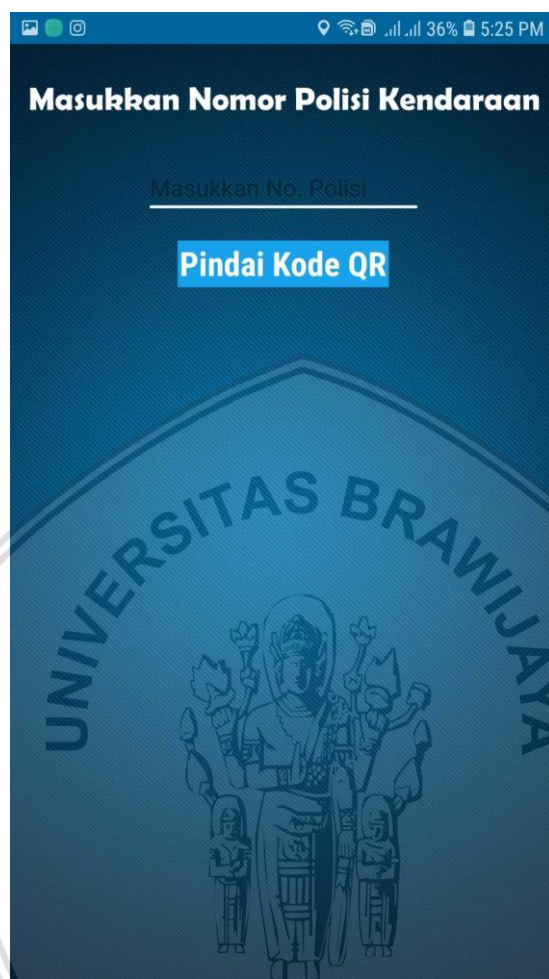
Gambar 5.11 berikut ini merupakan hasil implementasi dari perancangan menu utama.



**Gambar 5.11 Antarmuka Menu Utama**

## 2. Implementasi Antarmuka *Input* Nomor Polisi

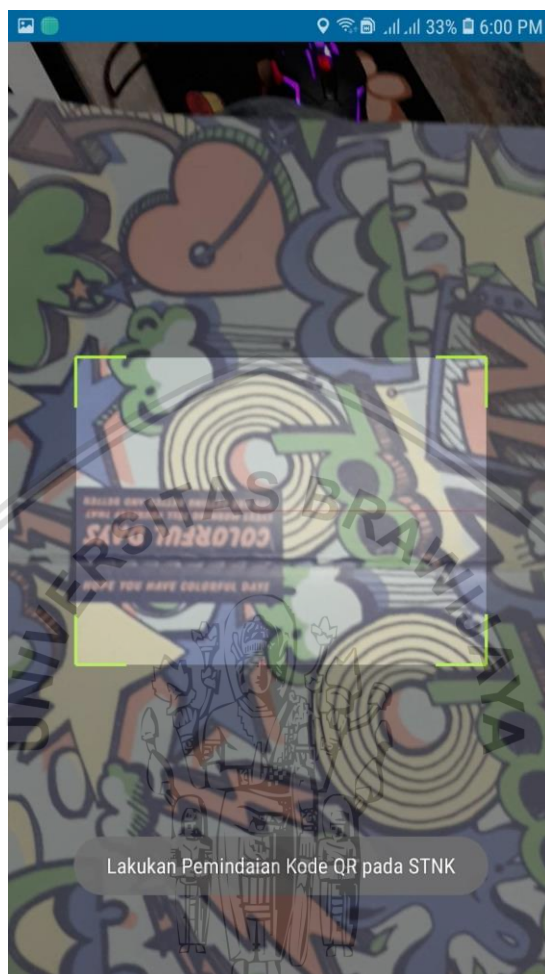
Gambar 5.12 berikut ini merupakan hasil implementasi dari perancangan *input* nomor polisi.



Gambar 5.12 Antarmuka *Input* Nomor Polisi

### 3. Implementasi Antarmuka Pindai Kode

Gambar 5.13 berikut ini merupakan hasil implementasi dari perancangan pindai kode.



**Gambar 5.13 Antarmuka Pindai Kode**



#### 4. Implementasi Antarmuka Informasi Pelanggaran

Gambar 5.14 berikut ini merupakan hasil implementasi dari perancangan informasi pelanggaran.

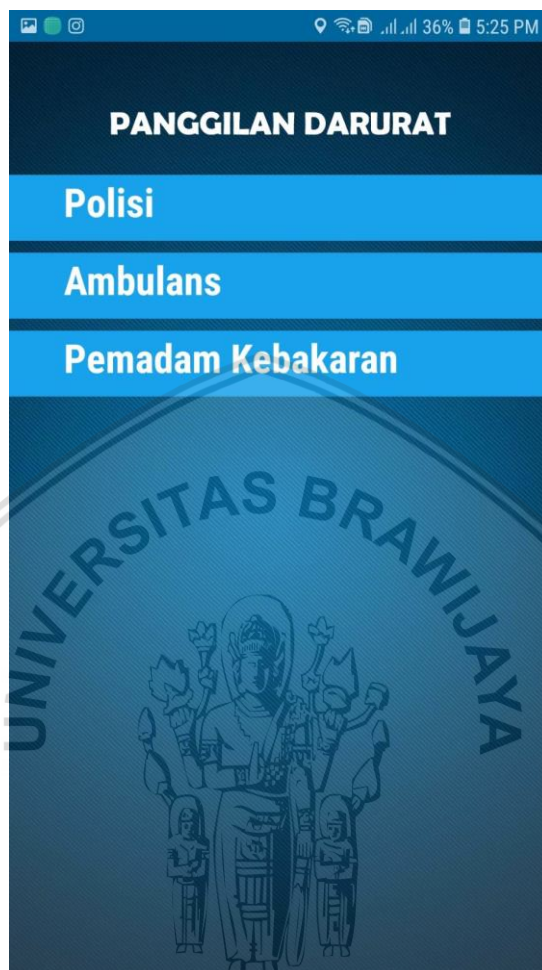


Gambar 5.14 Antarmuka Informasi Pelanggaran



## 5. Implementasi Antarmuka Panggilan Darurat

Gambar 5.15 berikut ini merupakan hasil implementasi dari perancangan panggilan darurat.



Gambar 5.15 Antarmuka Panggilan Darurat

## BAB 6 PENGUJIAN

Tahap pengujian dilakukan setelah sudah melakukan implementasi sistem. Pengujian bertujuan untuk memeriksa apakah implementasi sudah sesuai dengan analisis kebutuhan dan perancangan sistem atau tidak. Tahap pengujian yang akan dilakukan adalah pengujian validasi dan pengujian performa.

### 6.1 Pengujian Validasi

Pengujian validasi dilakukan untuk mengetahui bahwa fungsi-fungsi yang dibangun telah sesuai dengan kebutuhan fungsional. Dalam pengujian validasi dibuat kasus uji yang mengacu pada kebutuhan fungsional. Kasus uji dibuat untuk masing-masing kebutuhan fungsional. Hasil yang didapatkan dari pengujian validasi akan dianalisis untuk mendapatkan kesimpulan apakah aplikasi telah memenuhi kebutuhan fungsional yang telah dideskripsikan atau belum.

#### 6.1.1.1 Kasus Uji

Kasus uji dibutuhkan dalam pengujian validasi karena dapat digunakan untuk memberi prosedur pengujian agar sesuai dengan kebutuhan fungsional. Kasus uji dibuat untuk masing-masing kebutuhan fungsional yang telah di deskripsikan. Detail kasus uji dapat dilihat pada Tabel 6.1 hingga Tabel 6.4. Masing-masing kasus uji akan diberi nama dengan kombinasi singkatan, TEST, singkatan validasi, dan nomor urut kasus uji.

Kasus uji input merupakan kasus uji bagi pengguna untuk melakukan proses memasukkan nomor polisi kendaraan ke dalam sistem serta melakukan pemindaian kode QR. Kasus uji input dapat dilihat pada Tabel 6.1. Dalam tabel ini dijelaskan tujuan pengujian, prosedur, dan hasil yang diharapkan dalam pengujian.

**Tabel 6.1 Kasus Uji Memasukkan Nomor Polisi Kendaraan**

Nomor Kasus Uji	SPTL_TEST_VAL_1
Nama Kasus Uji	Login
Obyek Uji	[SPTL-F-001]
Tujuan Pengujian	Pengujian dilakukan dengan tujuan untuk masuk ke dalam sistem dan memastikan bahwa aplikasi dapat memberikan keluaran yang sesuai dengan kebutuhan fungsional.
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Membuka aplikasi</li> <li>2. Menekan tombol “Informasi Pelanggaran”</li> <li>3. Memasukkan nomor polisi kendaraan</li> </ol>

**Tabel 6.1 Kasus Uji Memasukkan Nomor Polisi Kendaraan (Lanjutan)**

	4. Menekan tombol “Pindai Kode QR” 5. Melakukan pemindaian kode
Hasil yang diharapkan	Aplikasi dapat menampilkan hasil pencarian informasi jenis pelanggaran dengan masukan nomor polisi kendaraan dan pemindaian kode QR yang dilakukan oleh pengguna.

Kasus uji melihat informasi pelanggaran dan estimasi denda merupakan kasus uji bagi pengguna untuk melihat informasi pelanggaran ke dalam sistem. Kasus uji input dapat dilihat pada Tabel 6.2. Dalam tabel ini dijelaskan tujuan pengujian, prosedur, dan hasil yang diharapkan dalam pengujian.

**Tabel 6.2 Melihat Informasi Pelanggaran Dan Estimasi Denda**

Nomor Kasus Uji	SPTL_TEST_VAL_2
Nama Kasus Uji	Melihat Informasi Pelanggaran Dan Estimasi Denda
Obyek Uji	[SPTL-F-002]
Tujuan Pengujian	Pengujian dilakukan dengan tujuan untuk mengetahui dan memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional melihat informasi pelanggaran dan estimasi denda.
Prosedur Pengujian	1. Membuka aplikasi 2. Memasukkan nomor polisi dan pemindaian kode QR.
Hasil yang diharapkan	Aplikasi dapat menampilkan hasil informasi pelanggaran seperti jenis pelanggaran, waktu pelanggaran, lokasi pelanggaran dan estimasi denda.

Kasus uji melihat notifikasi pemberitahuan mendekati *traffic light* merupakan kasus uji bagi pengguna untuk menampilkan notifikasi yang sesuai. Kasus uji input dapat dilihat pada Tabel 6.3. Dalam tabel ini dijelaskan tujuan pengujian, prosedur, dan hasil yang diharapkan dalam pengujian.

**Tabel 6.3 Melihat Notifikasi Pemberitahuan Mendekati *Traffic Light***

Nomor Kasus Uji	SPTL_TEST_VAL_3
Nama Kasus Uji	Melihat Notifikasi Mendekati <i>Traffic Light</i>
Obyek Uji	[SPTL-F-003]
Tujuan Pengujian	Pengujian dilakukan dengan tujuan untuk mengetahui dan memastikan bahwa aplikasi dapat menampilkan notifikasi mendekati <i>traffic light</i> dengan tepat.

**Tabel 6.3 Melihat Notifikasi Pemberitahuan Mendekati *Traffic Light* (Lanjutan)**

Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Membuka aplikasi</li> <li>2. Pengguna melewati daerah yang mendekati <i>traffic light</i></li> <li>3. Sistem menampilkan notifikasi</li> </ol>
Hasil yang diharapkan	Aplikasi dapat menampilkan notifikasi di titik yang sesuai dengan daerah rawan tilang dengan tepat.

Kasus uji melihat layanan keadaan darurat merupakan kasus uji bagi pengguna untuk melakukan panggilan darurat. Kasus uji melihat layanan keadaan darurat dapat dilihat pada Tabel 6.4. Dalam tabel ini dijelaskan tujuan pengujian, prosedur, dan hasil yang diharapkan dalam pengujian.

**Tabel 6.4 Melihat Layanan Keadaan Darurat**

Nomor Kasus Uji	SPTL_TEST_VAL_4
Nama Kasus Uji	Melihat Layanan Keadaan Darurat
Obyek Uji	[SPTL-F-004]
Tujuan Pengujian	Pengujian dilakukan dengan tujuan untuk mengetahui dan memastikan bahwa aplikasi dapat menampilkan <i>dial pad</i> yang sesuai.
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Membuka aplikasi</li> <li>2. Menekan tombol “Panggilan Darurat”</li> <li>3. Memilih dan menekan salah satu tombol, diantaranya : Polisi, Ambulans, dan Pemadam Kebakaran.</li> <li>4. Menampilkan <i>dial pad</i> sesuai tombol yang di tekan sebelumnya.</li> </ol>
Hasil yang diharapkan	Aplikasi dapat menampilkan nomor pada <i>dial pad</i> sesuai dengan tombol yang di tekan.

#### 6.1.1.2 Hasil Pengujian Validasi

Berdasarkan kasus uji yang telah dibuat dari Tabel 6.1 hingga Tabel 6.4, didapatkan hasil pengujian validasi yang dapat dilihat pada Tabel 6.5.

Tabel 6.5 Hasil Pengujian Validasi

Nomor Kasus Uji	Hasil yang Diharapkan	Status Validasi
SPTL_TEST_VAL_1	Aplikasi dapat menampilkan hasil pencarian informasi jenis pelanggaran dengan masukan nomor polisi kendaraan dan pemindaian kode QR yang dilakukan oleh pengguna.	  
		Valid

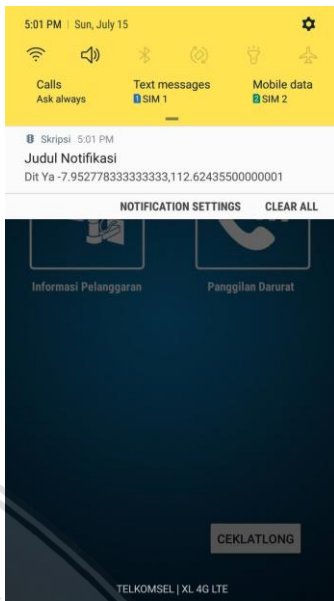
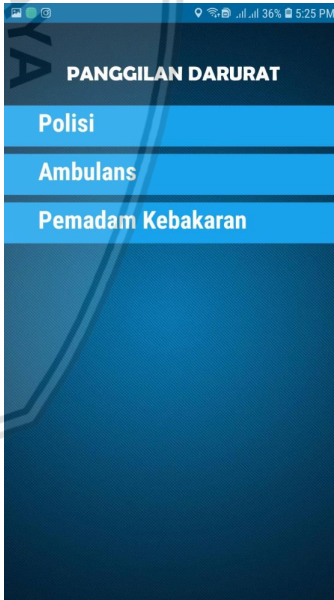


Tabel 6.5 Hasil Pengujian Validasi (Lanjutan)

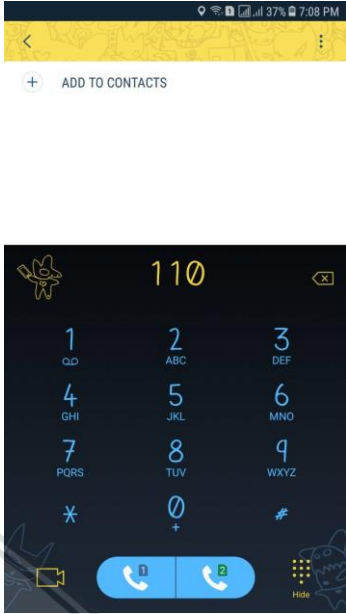
Nomor Kasus Uji	Hasil yang Diharapkan	Status Validasi
SPTL_TEST_VAL_2	Aplikasi dapat menampilkan hasil informasi pelanggaran seperti jenis pelanggaran, waktu pelanggaran, lokasi pelanggaran dan estimasi denda.	  
		Valid



Tabel 6.5 Hasil Pengujian Validasi (Lanjutan)

Nomor Kasus Uji	Hasil yang Diharapkan	Status Validasi
SPTL_TEST_VAL_3	Aplikasi dapat menampilkan notifikasi di titik yang sesuai mendekati <i>traffic light</i> dengan tepat.	 <p>Valid</p>
SPTL_TEST_VAL_4	Aplikasi dapat menampilkan nomor pada <i>dial pad</i> sesuai dengan tombol yang di tekan.	

Tabel 6.5 Hasil Pengujian Validasi (Lanjutan)

		
		Valid

## 6.2 Pengujian Performa

Pengujian performa dilakukan dengan melakukan tiga percobaan terhadap aplikasi. Pengujian ini dilakukan untuk melihat respon sistem aplikasi pemantau *traffic light* dengan parameter jarak dan kecepatan kendaraan yang telah dibangun. Mekanisme dalam pengujian performa ini dilakukan dengan melewati daerah *traffic light* dengan kecepatan kendaraan yang berbeda-beda, kemudian dilihat apakah notifikasi muncul di titik yang tepat dan sesuai. Jika sesuai, maka dapat diputuskan bahwa hasil munculnya notifikasi bernilai benar, jika notifikasi muncul di luar radius titik daerah rawan tilang, maka dapat diputuskan bahwa notifikasi bernilai salah. Berikut data hasil kasus uji yang dilakukan pada aplikasi dapat dilihat pada Tabel 6.6.

Tabel 6.6 Pengujian Performa

Pengujian Ke-	Kecepatan Kendaraan (km)	Jarak Radius (m)
1	20	84
2	40	40
3	60	22

## 6.3 Analisa Hasil Pengujian

Setelah semua pengujian telah selesai dilakukan, hasil dari semua pengujian dianalisis untuk dapat menarik kesimpulan bahwa hasil dari semua pengujian

yang telah selesai dilakukan telah memenuhi kebutuhan yang telah dijelaskan pada analisis kebutuhan.

### **6.3.1 Pengujian Validasi**

Dalam pengujian validasi semua fungsi telah diuji berdasarkan pengujian yang dilakukan dengan mencoba aplikasi yang dijalankan dan menjalankan aplikasi sesuai dengan prosedur pengujian yang sudah ditentukan, dari semua pengujian validasi tidak terdapat fungsi yang tidak valid yang berarti bahwa semua fungsi telah sesuai dengan kebutuhan yang telah ditentukan pada analisis kebutuhan.

### **6.3.2 Pengujian Performa**

Dalam pengujian performa dilakukan dengan menguji aplikasi untuk melihat respon dari suatu sistem dengan parameter jarak sejauh 100m dari titik lokasi uji dan kecepatan kendaraan. Dari pengujian performa dapat di ambil keputusan bahwa rata-rata dari ketiga percobaan tersebut dengan menggunakan kecepatan  $\pm 40\text{km/jam}$  dengan selisih jarak 48,6m serta aplikasi mampu berjalan dengan baik apabila terkoneksi dengan jaringan internet.



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian pengembangan aplikasi pemantau *traffic light* e-tilang kota Surabaya berbasis android, maka dapat disimpulkan sebagai berikut:

1. Berdasarkan hasil analisis dan pemodelan kebutuhan didapatkan seorang aktor yang memiliki empat kebutuhan fungsional dan satu kebutuhan non fungsional, kebutuhan fungsional tersebut dimodelkan dalam *use case diagram* serta *use case scenario*.
2. Hasil dari tahapan perancangan menghasilkan perancangan basis data yang digambarkan dengan *entity relationship diagram*, perancangan arsitektur yang di gambarkan dengan *sequence diagram* sebanyak empat kebutuhan fungsional dan lima *class diagram* yang terbentuk. Perancangan algoritma dibuat sebanyak dua algoritma dari fitur utama. Terakhir adalah perancangan *user interface* yang dibuat sebanyak lima perancangan dari kebutuhan fungsional.
3. Implementasi yang telah dilakukan menghasilkan implementasi basis data yang dikonversikan ke dalam JSON, implementasi *class diagram* menggunakan bahasa Java dengan *engine* Android Studio, implementasi *user interface* dilakukan dengan aplikasi Adobe Photoshop kemudian diimplementasikan kedalam *engine* Android Studio.
4. Pengujian dilakukan dengan *black box testing* untuk pengujian validasi kebutuhan fungsional dan pengujian performa dilakukan untuk memastikan performa dari fungsi sistem pemantau *traffic light*.

### 7.2 Saran

Dari penelitian yang sudah dilakukan pada pengembangan aplikasi *traffic light* e-tilang menggunakan Google *Geofencing* API berbasis android, berikut merupakan saran untuk penelitian berikutnya dengan topik yang sama :

1. Sistem dapat berjalan di latar belakang, tanpa membuka aplikasi sebelumnya.
2. Sistem dapat menghitung jumlah total denda pelanggaran yang telah dilanggar.
3. Dalam penelitian ini, sistem operasi Android yang digunakan adalah Marshmallow, untuk pengembangan penelitian selanjutnya dapat menggunakan sistem operasi Android Nougat untuk pembuatan aplikasi yang lebih kompatibel.

## DAFTAR PUSTAKA

- Garzon, S. R. & Deva, B., 2014. *Geofencing 2.0: Taking Location-based Notifications to the Next Level. UBICOMP.*
- Hadi, S. 2017. *Tilang CCTV Mampu Mengurangi Pelanggaran Lalu Lintas di Surabaya.* Tersedia di : <http://news.metrotvnews.com/read/2017/10/26/778715/tilang-cctv-mampu-mengurangi-pelanggaran-lalu-lintas-di-surabaya>. [Diakses : 20 Maret 2018]
- Herdi, H. 2014. *Belajar Membuat Aplikasi Android Menggunakan Android Studio.* Tersedia di: <http://www.twoh.co/2014/09/28/belajar-membuat-aplikasi-android-menggunakan-android-studio.html>. [Diakses : 20 Maret 2018]
- Larasati, W. A. 2017. *Pelanggaran Lalu Lintas di Kota Surabaya Menurun Setelah Tilang Lewat CCTV.* Tersedia di : <http://news.liputan6.com/read/3094520/pelanggaran-lalu-lintas-di-kota-surabaya-menurun-setelah-tilang-lewat-cctv>. [Diakses : 23 Maret 2018]
- Nugroho, A. 2008. *Pemrograman Java Menggunakan IDE Eclipse Callisto. Andi Offset.* Yogyakarta.
- Rachman, A. F. 2015. *Android Kuasai Asia Tenggara, di Indonesia Paling Juara.* Tersedia di : <https://inet.detik.com/consumer/d-3054169/android-kuasai-asia-tenggara-di-indonesia-paling-juara>. [Diakses : 24 Maret 2018]
- Ridwan, M. 2017. *Denda Pelanggaran Lalu Lintas di Surabaya Rp. 10 Miliar.* Tersedia di : <https://www.lensaindonesia.com/2017/12/28/denda-pelanggaran-lalu-lintas-di-surabaya-rp10-miliar.html>. [Diakses : 20 Maret 2018]
- Sommerville, Ian. 2011. *Software Engineering 9<sup>th</sup> edition.* Pearson Education, Inc.
- Trenggono, E. P., Sukmaaji, A., & Taufik, V. M. 2012. *Rancang Bangun Sistem Informasi Kontrol Kondisi Lalu Lintas Dengan Kamera Pemantau CCTV Berbasis GIS.* 7-10.
- Wahyudin. 2014. *Kendaraan di Surabaya Tambah 17 Ribu Lebih Sebulan.* Tersedia di : <http://www2.jawapos.com/baca/artikel/9796/kendaraan-di-surabaya-tambah-17-ribu-lebih-sebulan>. [Diakses : 20 Maret 2018]
- Widiarti, P. 2018. *Tahun Ini, CCTV e-Tilang di Surabaya Ditambah di 15 Titik.* Tersedia di : <http://kabar24.bisnis.com/read/20180104/78/723509/tahun-ini-cctv-e-tilang-surabaya-ditambah-di-15-titik>. [Diakses : 20 Maret 2018]

Wongsokuncoro, H. 2016. *Rancang Bangun Pendeteksian Pelanggaran Pada Traffic Light Berbasis Mikrokontroller.* 4-7.

